Conceptscope

Organizing and visualizing knowledge in documents based on domain ontology

Zhang, Xiaoyu; Chandrasegaran, Senthil; Ma, Kwan Liu

DOI
10.1145/3411764.3445396

Publication date
2021

Document Version
Final published version

Published in
CHI 2021 - Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems

ConceptScope: Organizing and Visualizing Knowledge in Documents based on Domain Ontology

Xiaoyu Zhang
xybzhang@ucdavis.edu
Department of Computer Science, University of California, Davis
Davis, California

Senthil Chandrasegaran
r.s.k.chandrasegaran@tudelft.nl
Faculty of Industrial Design Engineering, Delft University of Technology
2628 CE Delft, The Netherlands

Kwan-Liu Ma
klma@ucdavis.edu
Department of Computer Science, University of California, Davis
Davis, California

ABSTRACT

Current text visualization techniques typically provide overviews of document content and structure using intrinsic properties such as term frequencies, co-occurrences, and sentence structures. Such visualizations lack conceptual overviews incorporating domain-relevant knowledge, needed when examining documents such as research articles or technical reports. To address this shortcoming, we present ConceptScope, a technique that utilizes a domain ontology to represent the conceptual relationships in a document in the form of a Bubble Treemap visualization. Multiple coordinated views of document structure and concept hierarchy with text overviews further aid document analysis. ConceptScope facilitates exploration and comparison of single and multiple documents respectively. We demonstrate ConceptScope by visualizing research articles and transcripts of technical presentations in computer science. In a comparative study with DocuBurst, a popular document visualization tool, ConceptScope was found to be more informative in exploring and comparing domain-specific documents, but less so when it came to documents that spanned multiple disciplines.

CCS CONCEPTS

• **Human-centered computing → Information visualization**.

KEYWORDS

Visualization, Ontology, Knowledge Representation

ACM Reference Format:
Xiaoyu Zhang, Senthil Chandrasegaran, and Kwan-Liu Ma. 2021. ConceptScope: Organizing and Visualizing Knowledge in Documents based on Domain Ontology. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3411764.3445396

1 INTRODUCTION

Text visualization techniques have evolved as a response to the virtual explosion of text data available online in the last few decades. Specifically, they aim to provide a visual overview—what digital humanities now call "distant reading" [31]—of large documents or large collections of documents, and help the researcher, investigator, or analyst find text patterns within and between documents (e.g. [41]). Most of these visualization techniques are domain-independent and do not provide a knowledge-based overview of documents. There have been approaches to provide a visual overview of the semantic content of documents (e.g. [8]). Such approaches have typically looked to lexical hypernymy (is-a relationships) to provide a conceptual overview of the text.

However, when examining domain-specific documents such as research papers, medical reports, or legal documents, it is necessary to examine the documents from the point of view of that specific domain. For instance, when examining a research paper in computer science, a computer science researcher may be interested in whether the paper concerns a general overview of a subject, such as "computer graphics", or concerns more specific concepts such as "infographics" or "TreeMap visualizations". Similarly, the researcher may want to compare papers that appear in the same conference session to see the similarities and differences that may exist between the papers. In such scenarios, the overview visualizations should also represent the computer science domain and how the knowledge is structured in the domain.

While approaches such as topic modeling can provide a bottom-up categorization or thematic separation of a document's text, domain knowledge is often organized formally by experts in the corresponding domains using Ontologies. An ontology, defined as an "explicit specification of a conceptualization" [19, p. 199], is a widely-accepted way in which domain knowledge is formally represented. A knowledge-based overview of a document that uses as a reference the corresponding domain ontology can thus provide a conceptual overview for the domain expert. Such a view can also be used structurally to help the expert compare two or more documents based on the concepts they cover.

In order to aid document examination from the viewpoint of a specific domain, we present ConceptScope, a text visualization technique that provides a domain-specific overview by referring to a relevant ontology to infer the conceptual structure of the document(s) being examined. ConceptScope uses a Bubble Treemap view [17] to represent concept hierarchies, highlighting concepts from the ontology that exist within the document and their relationships with other concepts in the document, as well as key "parent" concepts in the Ontology. Each concept "bubble" is also populated with a word cloud that represents text from the document that relates to the concept, providing a contextual overview. Through a set of multiple coordinated views of text, structural overviews, and

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445396

keyword-in-context (KWIC) views, ConceptScope helps users navigate a document from a specific domain perspective. ConceptScope can also be used to visually and conceptually compare multiple documents using the same domain ontology as a reference. To aid a domain novice, we also provide the user with navigable tooltips that provide concept explanations that link to external references.

We illustrate the utility of ConceptScope by building a prototype application[1] that visualizes computer science-related documents such as research abstracts and articles using the Computer Science Ontology (CSO) as its reference. Through a set of use-case scenarios, we highlight the navigation, exploration, and comparison functions afforded by the technique, and discuss its extension to other domains and scenarios. We also present a brief comparison of ConceptScope with DocuBurst [8] through a qualitative, between-subjects study. Based on our observations, we find that ConceptScope's ontology-based visualization and its grouping of concept-related word clouds in the Bubble Treemap helps participants define and contextualize concepts, and explore new concepts related to a given concept. However, ConceptScope's domain-dependency makes it less suitable for viewing and comparing documents that span domains.

2 RELATED WORK

This paper proposes an interactive knowledge-based overview representation of text content. For our approach, we draw from existing techniques to identify themes or topics in the text, and visual representations of these topics. In this section, we outline existing work in this area and explain our reasoning behind our choice of inspiration from the existing work.

2.1 Thematic Visualizations of Document Content

Initial approaches to providing overview visualizations of document content used metrics such as sentence length, Simpson's Index, and *Hapax Legomena* as "literature fingerprints" to characterize documents [26]. This approach was later used to create a visual analysis tool called VisRA [34] that helped writers review and edit their work for better readability using these representations. Among less abstract representations, Wordle [45] is the most popular. Wordle represents a text corpus as a cluster of words called a *word cloud*, with each word scaled according to its frequency of occurrence in the text. This idea is adapted to other techniques to characterize document content and structures within text, such as the Word Tree [48], which aggregated similar phrases in sentences in a text, Phrase Nets [44] that visualized text as a graph of concepts linked by relationships of the same type found in the text, and Parallel Tag Clouds [9], that show tag clouds on parallel axes to compare multiple documents.

When examining multiple text documents, it is important to identify the various types of connections between them. One of the most well-known tools used to identify inter-document connections is Jigsaw [41], which uses names, locations, and dates to show list, calendar, and thumbnail views of multiple documents. While Jigsaw simply uses text occurrences to form the connections, more sophisticated approaches have since been proposed. Tiara [49]—another system designed for intelligence analysis—uses topic modeling with a temporal component to highlight the change in document themes over time. ThemeDelta [15] allows thematic comparison between multiple documents (or similar documents over time) by combining word clouds with parallel axis visualizations.

More recently, topic modeling-based approaches have been incorporated to provide thematic overviews of text content. For instance, TopicNets [18] uses a graph-based representation where both documents and topics are nodes and links exist between documents and topics, thus serving to form clusters of thematically-related documents. Serendip [2] refines this idea and provides a multi-scale view of text corpora. It uses topic modeling along with document metadata to view patterns at the corpus level, text level, and word level. Oelke et al. [35] use a topic model-based approach to compare document collections, using what they call a "DiTop-View" with topic glyphs arranged on a 2D space to represent the document distribution. ConToVi [12] is a more recent work that uses topic modeling on multi-party conversations to reveal speech patterns of individual speakers and trends in conversations. While topic model-based approaches are useful for identifying themes within collections of documents, a knowledge-based approach requires the use of human-organized representations of information, which are discussed in the following section.

2.2 Knowledge-Based Visualizations

As structured knowledge representation models [16], ontologies are widely used in the field of medicine/biology [16], engineering [36, 51], sociology [22], computer science [42], and so on. Achich et al. [1] review different application domains and generic visualization pipelines of ontology visualization.

According to various application fields and utilizing purpose, there are multiple methods to visualize the knowledge stored in an ontology. The review of Katifori and Akrivi [25] systematically categorized these methods according to the dimension of the visualization. Ten years later, Dudáš et al. [11] further extended this work by adding more recently emerged visualizations. Among these visual encodings, we find inspiration in the matrix view of NodeTrix [22], the sunburst view of PhenoBlocks [16], and the context view of NEREx [13].

Our work is inspired by DocuBurst [8], which was the first visualization from the point of view of a human-organized structure of knowledge. DocuBurst uses hyponymy, or "is-A" relationship in the English lexicon to identify hierarchical relationships within a given document, or when comparing two documents. The hierarchy is visualized as a sunburst diagram supported by coordinated views of text content and keyword-in-context views. While DocuBurst uses WordNet—a lexical database of the English language—as its reference, we use domain ontologies as ours, in order to provide a more focused, domain-specific overview of documents.

2.3 Hierarchical Layouts

Visualization of a knowledge-based document overview needs to incorporate the hierarchical information inherent to the knowledge base. While a tree is the common representation of such a hierarchy, it is usually more suitable for showing the structure rather than the content of the information presented. The most famous alternative

[1]The source code of our prototype system is available at https://github.com/Xiaoyu1993/ConceptScope/

for representing hierarchical information is the TreeMap [38], a two-dimensional, space-filling layout that represents hierarchy through nesting and a second quantity such as percentage contribution to the whole as the area. Alternatives to TreeMaps such as Icicle plots and Radial TreeMaps [4] and Sunburst diagrams [40] have since been proposed and incorporated into standard visualizations of hierarchies. DocuBurst [8] referenced in the previous section uses the Sunburst diagram as its hierarchical visualization.

While the original TreeMap has afforded enough space in the representation to portray content, it often comes at the cost of some loss of detail in the hierarchy. Alternatives such as circle packing [47] and more recently, Bubble Treemaps [17] have been proposed to address this issue. We incorporate the Bubble Treemap into our design for its relative compactness compared to circle packing, and its use of space that allows for some content representation.

3 REQUIREMENTS AND DESIGN

In this section, we break down our overall need to provide a knowledge-based overview of document content into specific requirements to inform the design of ConceptScope. We apply Collins et al.'s [9] question "What is this document about?" to the general "distant reading" tasks for both single text analysis and parallel text analysis posed in Jänicke et al. [24], typically addressed using intrinsic text properties such as entity/location occurrences, text frequencies, etc., but not domain knowledge. Our requirements stem from exploring tasks of hierarchical overview, document comparison, and concept exploration using a knowledge base as reference.

R1 Provide Conceptual Overview: When reading a long document from an unfamiliar domain—such as an academic paper—the reader can benefit from a high-level overview of the information provided. While word clouds can provide a simple overview of the *text* in the document, a lack of understanding of the technical terms might hinder the reader in understanding the overview representation. Instead, an overview that stems from a fundamental categorization of the domain itself—as represented by the hierarchical organization of concepts often available in an ontology—can provide an overview that is accessible to both novices and experts in the domain.

R2 Reveal Contextual Information: The document text and the ontology do not always overlap. From the point of view of the ontology, the document contains non-relevant information, but information nevertheless important for the reader. For instance, a research paper introducing a new search algorithm can introduce several concepts in the knowledge base of search algorithms. The paper would also make arguments for and against certain algorithms. The reader may benefit considerably from the structure and content of these arguments, which are lost if the overview visualization focuses solely on the ontological components. A way to provide the contextual information surrounding these concepts is thus needed.

R3 Support Exploration of New Knowledge: When exploring a concept that is a sub-domain of a domain that is only partially known to the reader, they may be interested in other sub-domains of the domain. For example, if the term "quicksort" appears in an algorithm paper, the reader might want to know of other sorting algorithms such as "bubble sort" and "merge sort". They may also want to learn about related terms such as "divide and conquer" and "time complexity". These new terms may not appear in the document text, but forms an essential component of knowledge that extends from—and aids the understanding of—the core concept (i.e. quicksort). We thus need ways to enable users to access information from the ontology that is related to the concept of interest.

R4 Support Multi-document Comparison: Document comparison is a common requirement that emerges from the creation of visual overviews of documents [8]. In the case of our scenario, the comparison is likely to be conceptual: to get a quick comparison of concepts that are common to multiple documents, and those that are unique to one. The reader may also want to simply compare the differences between the information provided in two documents. While documents such as academic papers may contain an abstract that summarizes the main content of the article, it may not be sufficient enough to cover all the concepts that are covered in the papers, not to mention the similarities and differences. Therefore, our tool should be able to provide visual support for users to compare and analyze the conceptual structure and content between two or more documents.

4 IMPLEMENTATION

In order to provide the knowledge-based conceptual overviews of a given document, an appropriate mechanism is needed to parse the document and compose queries to the reference ontology. An appropriate representation of the concept needs to be automatically generated in a way that reflects its hierarchy in the domain ontology as well as its occurrence in the document. To achieve this, we need to incorporate techniques from multiple areas including natural language processing, ontology querying, and information visualization. Fig. 1 shows the framework of assembling them into a pipeline and the section number describing the corresponding technical details.

4.1 Generating Query Candidates

Ontology queries are typically performed using SPARQL (SPARQL Protocol And RDF Query Language) [46], which typically use "triples" (subject, predicate, and object) or parts thereof. In our case, trials showed that an exact triple was unlikely to be constructed from the document, nor was it deemed necessary. Instead, it was more important to have the subjects or objects be specific terms that are likely to be present in the ontology. We construct these queries from the document with a sentence-level granularity. In order to construct the query terms, we use two approaches: noun chunking, and n-gram identification.

Noun chunking is the process of extracting subsets of noun phrases such that they do not contain other noun phrases within them [6]. This allows us to identify specific terms that may be relevant to a domain ontology. For instance, when referencing the computer science ontology, terms such as "object-oriented programming" and "local area network" are much more meaningful than

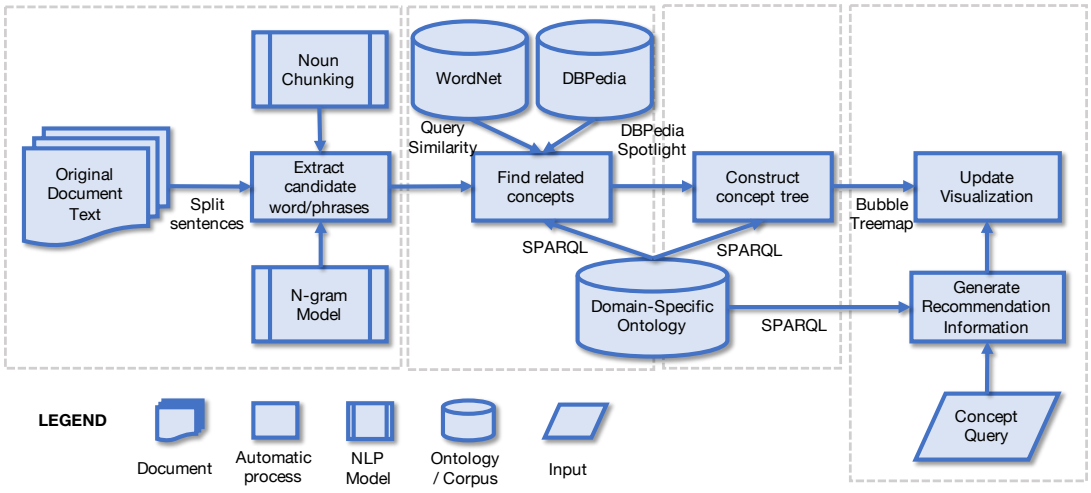


Figure 1: Data processing pipeline for ConceptScope.

the individual words that make up these terms ("local", "object", or "area"). For this reason, we also do not resort to stemming or lemmatization as they change the morphology of the word (e.g., "oriented", if lemmatized to "orient", forms "object-orient programming") which renders the noun chunk invalid as a query candidate. Noun chunks can also include leading or trailing stop words, which are trimmed in order to generate the query candidates.

Noun chunking can produce phrases that contain query candidates but are not query candidates themselves. For instance, a paper about animation may include multiple variances of animation like "2D computer animation", "stop-motion animation" and "animated transition". Some of these may appear within noun chunks, but not by themselves. To identify such cases, we identify groups of words that commonly occur together in the document as n-grams.

4.2 Mapping Queries to Concepts

Once the query candidates are identified, the next step is to map these candidates to the corresponding concepts in the domain ontology of interest. This involves two steps: (1) perform identical matches, i.e. concepts that correspond exactly to those in the ontology, and (2) reduce the number of "failed" matches, i.e. concepts that are related but not present in the ontology. Step 2 is often necessary as domain ontologies are not all uniformly mature. For instance, Computer Science Ontology is not as well-populated as, say, medical or biological ontologies such as the human phenotype ontology.

The two steps—accurate matching and fuzzy matching—are illustrated in lines 8 through 15 in Algorithm 1. For any given candidate, we first look for an accurate match in the domain-specific ontology. We then construct a dictionary that includes all of the concepts in the ontology for an effective search. However, the number of concepts that can be directly detected by accurate matching is small. This is because of the mismatch between specific forms in which a concept is listed in the ontology and its many variations in the document. For instance, "object-oriented programming" may be the exact match in the ontology, but it might appear in the text as "object-oriented approach" which is clearly related but cannot be identified with an accurate match. In order to solve this problem, we introduce a fuzzy match.

Algorithm 1 Detect CSO Concepts in Document

Input: document text *stringDoc*
Output: concept dictionary *dictConcept*

1: $listSent \leftarrow Split(stringDoc)$
2: $modelNGram \leftarrow TrainNGram(listSent)$
3: $dictConcept \leftarrow \emptyset$
4: **for** *stringSent* **in** *listSent*: // iterate over each sentence of the document
5: $listNGram \leftarrow modelNGram(stringSent)$ // identify initial query terms
6: $listChunk \leftarrow NounChunking(stringSent)$ // identify additional query terms
7: $listCand \leftarrow listChunk \cup listNGram$
8: **for** *stringCand* **in** *listCand*: // iterate over each candidate query term
9: **if** $QueryCSO(stringCand) \neq \emptyset$: // accurate matching
10: $dictConcept \leftarrow dictConcept \cup QueryCSO(stringCand)$
11: **else**: // fuzzy matching
12: $fuzzyCand \leftarrow DBpediaSpotlight(stringCand)$ // get candidate DBpedia concepts
13: $fuzzyCand \leftarrow Filter(fuzzyCand, threshold)$ // filter candidate DBpedia concepts according to similarity
14: **if** $QueryCSO(fuzzyCand) \neq \emptyset$: // link the filtered DBpedia concepts back to CSO concepts
15: $dictConcept \leftarrow dictConcept \cup QueryCSO(fuzzyCand)$

The goal of fuzzy matching is to match the candidate to a concept that is very close to but not exactly equal to the candidate. In our prototype system, we use the computer science ontology (CSO) as the domain-specific ontology. The CSO also incorporates links of the form *"sameAs"* (http://www.w3.org/2002/07/owl#sameAs), that connect to DBPedia [29], a broader, but less strictly-defined and less domain-specific ontology. We use these links and leverage the DBpedia Lookup Service [21] to find related DBpedia concepts and link them back to CSO. After checking the semantic similarity between the CSO concept detected in this way and the original

candidate query term using the Wu-Palmer similarity measure offered as a default function in WordNet [14], we add the concept to the dictionary if that similarity is above a threshold. This threshold is currently determined by trial and error.

4.3 Hierarchy Reconstruction

The concept dictionary constructed thus far does not yet incorporate hierarchical information. In order to retrieve and store the hierarchical information from the ontology, we query the paths from every detected concept to the root of the ontology and use them to restructure the concept dictionary as a tree. The final output of this algorithm—the concept tree—can be directly converted to a JSON file and used to automatically render the visualization.

5 CONCEPTSCOPE INTERFACE

In this section, we discuss the visualization design and the interactions supported in ConceptScope.

5.1 Visual Encoding

We choose Bubble Treemaps proposed by Görtler et al. [17] as our primary visualization. This visualization is originally designed for uncertainty visualization, but we find it suitable for our application in terms of hierarchy representation and space organization (**R1**). We use the original layout algorithm of the Bubble Treemap, but adapt the visual encoding and interaction strategies to meet our design requirements.

5.1.1 Hierarchy Presentation. In a Bubble Treemap, the deepest levels of the hierarchy are represented as circles, with successively higher levels forming contours around their "child" levels. We use the circles to represent the terms that appear (or have corresponding synonyms) in the original document as well as in the ontology. The outer contours represent concepts that do not explicitly appear in the document but still represent parent concepts from the ontology. These parent concepts are identified using the ontology query process demonstrated in Algorithm 1. The outermost contour forms the "root" of the ontology, with successive inner contours representing its child concepts. For example, in the computer science ontology (CSO) [37] we use for our case studies, the term "computer science" is the root concept in the ontology.

Inner Circles. The function of the innermost circles—representing concepts that are present in the ontology *and* in the document—is to provide a clear representation of the terms that are directly connected to the document. The size of the circles is proportional to the frequency with which the corresponding term appears in the document. The fill color of a given circle corresponds to the highest "parent concept" it belongs to, just below the root. Although the Bubble Treemap layout already gathers together circles that share the same parents, we visually reinforce such relationships by assigning the same color to circles with the highest common ancestor (besides the root). These "highest parent concepts", divide the root term into several subclasses and help users to better grasp the various areas the document covers. In order to make sure the circles' colors are perceptually uniform, we create the isoluminant palette [27] from the CIELAB color space to ensure perceptual uniformity between the concepts shown.

Surrounding Contours. The contours surrounding the circles show hierarchical relationships between the concepts that occur in the document. After exploring several encoding options for the contours to best represent related concepts while highlighting hierarchies, we chose fill colors of decreasing luminance to represent "deeper" contours in the hierarchy.

5.1.2 List Presentation. Effective as the Bubble Treemap is, it is not intuitive enough for the users to understand and grasp all necessary information at a glance (**R1**). Therefore, we augment the visualization with a multi-function widget (Fig. 2 (e)) which combines concept list, legend, and bar charts representing term frequencies to solve this problem. Inspired by scented widgets [50], the multi-function widget presents important supporting information in a compact representation. As a concept list, this tool represents every concept detected in the currently-loaded document(s) as a list item, the background color of which is the same as the corresponding concept circle(s) shown in the Bubble Treemap. We group the concepts sharing the "highest super topic" together, with an additional list item showing the common "highest super topic" of each group. This concept list also acts as a legend showing the connection between each color and their corresponding "highest super topic". We also attach a sparkline for each list item to show the distribution of current concept across multiple documents (when multiple documents are loaded).

5.1.3 Incorporating Word Clouds. An unlabeled Bubble Treemap can be too abstract a representation for the user to comprehend. On the other hand, labeling every concept may result in a cluttered view which would also make comprehension difficult. We thus provide three levels of labeling for the concept: unlabeled (if the concept circle is too small), labeled (if the concept circle is large enough to fit its corresponding concept name), and labeled with context (where a word cloud of related terms from the document is combined with the concept label) (**R2**). The interactions to control these views are discussed in the following section.

5.2 Interaction

ConceptScope provides linking between views and semantic overview and detail views to help analyze the document(s) and its concepts. These interactions support two modes of document analysis: exploration and comparison. We will first describe the overview and detail interactions and follow them with the modes of analysis.

5.2.1 Overview+Detail Interactions. To eliminate the potential confusion caused by the users' unfamiliarity with the Bubble Treemap, we introduce interactions to acquaint them with the visual schema and provide details on demand [39]. The Bubble Treemap provides a compact view of the domain-relevant concepts, their hierarchical structure in the ontology, as well as their context in the original document. In order to make this compact representation easier to understand, we design two interactions to present information that the user may seek: (1) a *level slicer* to "slice" the Bubble Treemap at any level to examine parent concepts, and (2) *semantic zooming*, which allows the user to zoom in to a concept circle to examine its corresponding word cloud (described in Sec. 5.1.3). The users can choose and combine these two tools according to their preference.

The **Level Slicer** is designed to help novice users quickly build a connection between the nested layout of the Bubble Treemap and the hierarchical structure of ontology (**R2, R3**). This tool allows the user to choose the level of the parent concept that they want to see

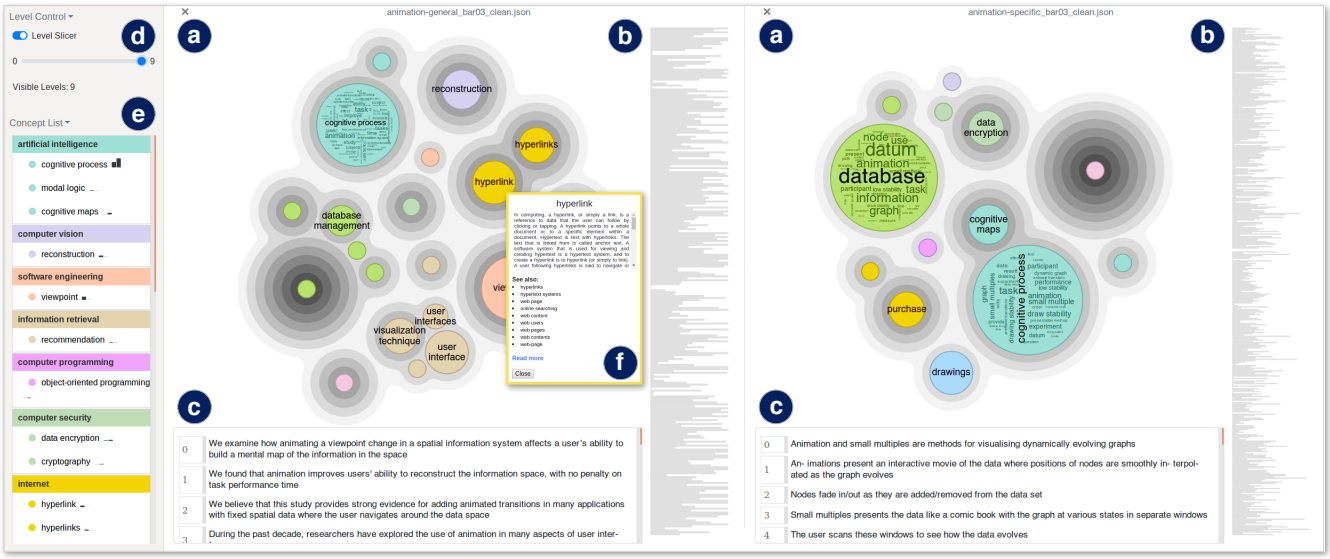


Figure 2: The ConceptScope interface representing two research papers discussing animation. The Bubble Treemaps (a) provide overviews, with the one on right showing a paper covering more specific topics than the one on the left. Supporting transcript (b) and text (c) views, along with a level slicer (d) and a list presentation (e) allow exploration and comparison between the documents, while a tooltip (f) allows examination of a concept of interest.

on the screen by sliding the slider bar. When the view initializes, all levels of the Bubble Treemap are shown to provide an overview, but the labels corresponding to parent contours are concealed. Once the "child" concepts are sliced away by the slicer, the corresponding labels of the newly exposed parent concepts are made visible. This tool facilitates users to inspect any cross section from the whole hierarchical structure that interests them.

Semantic Zooming is designed to provide different granularities of information based on users' needs (**R2, R3**). As explained in Sec. 5.1.3, users may see three levels of detail for the same concept circle: *unlabeled*, *labeled*, and *labeled with word cloud*. When users zoom in and out of the graph, the size of every circle changes and its appearance transforms among the three based on the available space inside it.

ConceptScope also reveals more information about a concept including its thumbnail, definition, related concepts, and its context in the text. These views allow the exploration of concepts that do not themselves occur in the document but are related to the ones that do occur (**R3**).

5.2.2 Exploration Mode. The exploration mode—meant for inspecting a single document—provides conceptual overview+detail representations of the document using the ontology as a reference. With the static Bubble Treemap, it is almost impossible for novice users to build the connection between a circle in the graph and a word/phrase in the original text. Users might want to explore related knowledge in the domain ontology about the concepts shown in the Bubble Treemap. Following the information-seeking mantra [39], we design a set of small widgets that can be easily evoked and interacted with to the Bubble Treemap.

To *connect the Bubble Treemap and the original document* (**R2**), we create a high-level transcript view and a raw text view. The high-level transcript view can be seen as a "minimap" of the document, with each sentence represented by a series of horizontal lines scaled to sentence lengths (Fig. 2 (b)). In the raw text view, the raw text is shown to provide a convenient context acquisition (Fig. 2 (c)). These two views as well as the Bubble Treemap view are fully coordinated, so that interacting with one view highlights related information in the other views. For example, if the users hover over a circle representing a concept in the Bubble Treemap view, the lines corresponding to the sentences that contain this concept in the transcript view and the text of the sentence in the raw text view are also be highlighted.

Interacting with a concept circle also reveals a tooltip that shows the concept definition, a link to the relevant concept page on DBPedia (**R3**)(Fig. 2(f)), and a thumbnail (if available in the corresponding DBPedia entry). The tooltip also provides links to other related concepts that may not be present in the document, to provide context from an ontology point of view.

5.2.3 Comparative Mode. The comparative mode assists users in comparing multiple documents and explore conceptual similarities and differences between the documents (**R4**). As the name suggests, loading multiple documents creates multiple, side-by-side Bubble Treemap views, one for each document. Concepts common to two or more documents are encoded in the same color across the Bubble Treemaps.

The comparative mode provides similar interactions as the exploration mode. In addition, the sparklines mentioned in Sec. 5.1.2 can provide users with a quick overview of the relative frequency with which each concept occurs across the documents. The users can compare the concepts that interest them by hovering or searching. If they know where a concept is located in any of the Bubble Treemaps, the user can simply hover on the corresponding circle or contour, which highlights the concept—if available—across all the Bubble Treemaps. They can also directly search for the concept in

the search field (top right corner in Fig. 2) to highlight all relevant circles and contours across the Bubble Treemaps. The users can thus quickly get an idea about where and how their concepts of interest are distributed across different documents.

The switchover between exploration mode and comparative mode does not require explicit user operation. Loading a single document shows the exploration mode while loading additional documents sets ConceptScope to comparison mode. The exploratory features are always available regardless of the number of documents, as comparison also requires a degree of exploration. We also provide a "switch" alongside the level slicer in Fig. 2(d) for semantic zooming to make sure the users can explore or compare the Bubble Treemap(s) at whatever number of levels and size they want.

6 USE-CASE SCENARIOS

We briefly illustrate the use of ConceptScope for exploring and comparing documents with two use-case scenarios: exploring an academic paper and comparing the transcripts of three TED talks.

6.1 Exploring an Academic Paper

We first use ConceptScope to visualize an academic paper [10] on automatic infographics generation, published in IEEE VIS 2019. To ensure the accuracy of our natural language processing components, we only keep the natural-language parts of the original paper and remove text in references, tables, formulae, and figure labels. We use the computer science ontology (CSO) as the reference ontology for this paper. Fig. 3 shows the visualization, with the same paper shown in DocuBurst [8] for reference.

The Bubble Treemap shows over 30 computer science concepts directly or indirectly mentioned in the paper (requirement **R1**). Inspecting the concept list on the left, we see that the highest parent concepts of the ones identified in the document range from "human-computer interaction" to "artificial intelligence" to "computer system". Zooming in, we click on the bubble representing "OCR" and a tooltip pops up with the definition of this concept as well as the recommendation of concepts related to this one (**R3**). We examine the definitions and where the concept appears in the word cloud to see that it points to the use of OCR to identify key text in existing infographics (**R2**). We also see that these and most concepts under "artificial intelligence" appear under the related work section. We thus infer that these concepts might only be mentioned as background or references to other work, and not as a fundamental contribution of the paper.

Figure 3 (right) shows the DocuBurst visualization using the root "message". We notice that almost all computer-science-related concepts identified by DocuBurst can be detected by ConceptScope as well. In terms of space efficiency, DocuBurst has the advantage of providing a more compact visualization with its Sunburst diagram. However, DocuBurst offers fewer options for contextual views. In ConceptScope, the word clouds in each concept circle provide a contextual overview and aid concept exploration outside the realm of the document with our detail-info tooltip of concepts and the links to DBPedia.

6.2 Comparing Transcripts of TED Talks

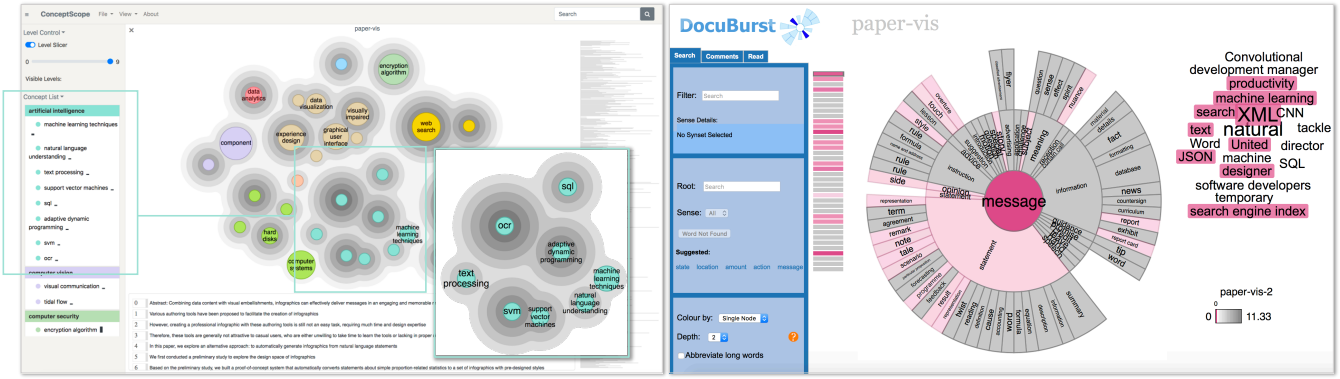
To illustrate multi-document comparison, we load the transcripts of three TED Talks [7, 23, 30], all of which are tagged under the "computers" category on the TED webpage. Fig. 4 shows the distribution and depth of concepts, along with information about each talk.

Loading all three documents into ConceptScope creates three panels (similar to that shown for two papers in Fig. 2), each containing the Bubble Treemap view, transcript view, and raw text view for the corresponding transcript. The Bubble Treemap immediately illustrates the differences and similarities between concepts across the three talks, which can further be explored as all three views are coordinated. We notice that all three of the talks mention concepts under the parent topics of "internet", "computer security" and "artificial intelligence". One reasonable explanation is that these topics cover many basic terms in computer science, so it is almost unavoidable to use them in a computer-science-related technical presentation. When inspecting the concept list and Bubble Treemaps, we notice that concepts that belong to "artificial intelligence" appear more in talk No. 2 and talk No. 3, which makes sense as the two talks have the additional tag of "AI" on the TED webpage.

Talk No. 1 discusses the issue of privacy on online forums, and concepts of privacy and anonymity fall outside the current version of the computer science ontology. In addition, the talk does not delve deep into computer science concepts. This results in a Bubble Treemap that covers very few concepts. Talk No. 2 is delivered by a data scientist who talks about computer science concepts, specifically "algorithms", "machine learning", and "deep learning", which are reflected in the Bubble Treemap. Finally, Talk No. 3 is presented by a philosopher who talks about broader implications of machine learning, also providing a historical perspective. This is reflected in the Bubble Treemap, showing the broadest concept coverage of the three talks, with no one concept being too dominant.

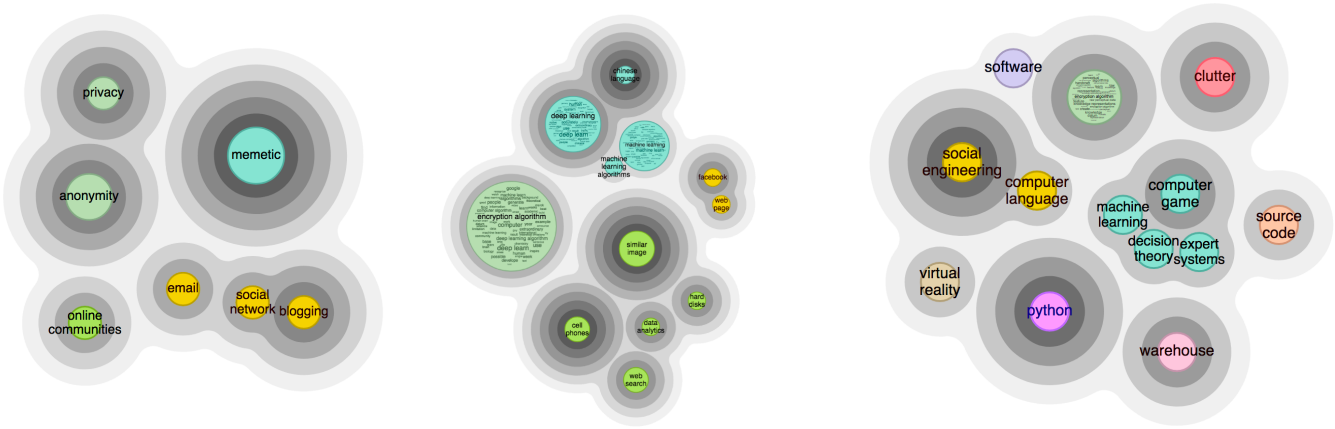
7 STUDY

We conducted a controlled study to evaluate whether the visualization & interaction design and the use of a domain-specific reference ontology renders ConceptScope effective in exploring single documents or comparing multiple documents. Specifically, we intended to understand whether ConceptScope was effective in helping users: (1) summarize the content of a document with a domain-specific concept overview (**R1**); (2) glean what a document says about any given concept in the context of the document (**R2**); (3) become aware of new concepts and their connections (**R3**); and (4) discover enough similarities and differences among multiple documents (**R4**). In order to provide a baseline, we used DocuBurst [8], the popular content-oriented document visualization tool that provides a non-domain-specific overview of documents using the WordNet [14] taxonomy. We thus conducted a between-subjects study comparing participants that used ConceptScope with participants that used DocuBurst. Please note that the generalizability of this study might be affected by the limited number of participants we could recruit and the diverse devices they used due to safety measures surrounding COVID-19. However, the way we report the insights were mainly based on patterns and not numbers, so the validity is not highly impacted by those factors.



(a) ConceptScope Interface (b) DocuBurst Interface

Figure 3: Overview of an IEEE VIS 2019 Paper [10] using ConceptScope (left) and using DocuBurst (right).



The case for anonymity online

Speaker Profile: Organization founder
Tag: Internet origin stories
Duration: 11:09
# of Concepts: 7

The wonderful and terrifying implications of computers that can learn

Speaker Profile: Philosopher
Tag: Artificial intelligence
Duration: 19:17
# of Concepts: 12

What happens when our computer gets smarter than we are?

Speaker Profile: Data Scientist
Tag: Artificial intelligence
Duration: 16:03
# of Concepts: 13

Figure 4: ConceptScope visualizations comparing the transcripts of three TED Talks. The title of each talk is shown in red under each visualization, along with speaker profile and talk metadata. Also shown is the number of concepts from CSO found in each document.

7.1 Participants

We recruited 18 participants (10 female, 8 male) aged between 18 and 44 years. The participants comprised 16 Ph.D. students, 1 undergraduate student, and 1 employee of a technology company. Seventeen participants had computer science backgrounds, of which 12 specialized in visualization and HCI, 1 in high-performance computing, 1 in natural language generation and multi-modal learning, while 3 didn't report their specialized field. The one remaining participant had a design and education background, specializing in learning and user experience design. Two of the 18 participants reported themselves as native English speakers.

7.2 Conditions and Task Design

Most document visualization systems use either intrinsic statistical information such as topic models and word co-occurrences, or human-curated categories that do not scale to large knowledge bases (e.g. [33]). Per Kucher et al.'s survey [28], which is currently up to date[2], DocuBurst is the only knowledge-based document exploration system. We thus chose DocuBurst as the baseline for our evaluation.

DocuBurst provides an overview of documents based on the non-domain-specific "is-a" relationship in WordNet, while our prototype is based on domain-specific ontologies, in this case, the Computer Science Ontology (CSO). We asked each participant to perform the

[2]Text Visualization Browser: https://textvis.lnu.se/

same tasks using the interface assigned to them (ConceptScope or DocuBurst) and compared interaction and behavior patterns across participants. Participants were given time to familiarize themselves with their assigned interface. They were then asked to perform the following tasks:

T1 Explore one single document: This task was divided into several sub-tasks, each aligned with a corresponding design requirement: (1) summarize the documents and provide relevant keywords (**R1**); (2) describe a specified concept based on its usage in the document (**R2**); (3) select (from a list of description) the context in which a given concept is used in the document (**R2**); (4) define several concepts before and after using the system, as well as rate confidence with the definition (**R3**); (5) identify concepts in the document related to a given concept (**R3**); and (6) list the concepts (that the participants did not know before the study) in the document (**R3**). Participants were also asked whether they read the documents before the study to account for potential confounds.

T2 Compare two documents: The participants were asked to compare two documents at a conceptual level (**R4**). Therefore, they were asked to identify common and unique concepts, as well as overall similarities and differences between the two documents. Again, they were asked whether they read the documents before the study to eliminate bias.

T3 Compare three documents: The questions that participants were asked to answer in this task were generally the same as task T2 but for three documents. One difference was that participants were suggested to "identify a theme and explain their difference within the theme" when identifying the difference between three documents. Since DocuBurst was not capable of comparing more than two documents, this task was only assigned to participants using ConceptScope in the study.

In order to mirror participants' regular reading experience, we chose computer-science related academic papers or technical reports for all tasks of this study. For task T1 we used Munzner's nested model for validating visualizations [32]. Task T2 involved two papers discussing animation techniques: the first, a general evaluation of how animation could help users build a mental map of spatial information [5], while the other focused on the role of animation in dynamic graph visualization [3]. To alleviate participant fatigue and manage their time, we chose to use relatively shorter transcripts of three 15–20 minute Ted Talks [7, 23, 43] in the "artificial intelligence" playlist instead of academic papers for task T3.

7.3 Study Setup

We conducted the study remotely owing to safety measures surrounding COVID-19. The participants were asked to access either of the tools from a remote server and participate in the study with their own machine and external devices. Fourteen of them used laptops with screen sizes ranging from 13 in. to 16 in. The other 5 used monitors with screen sizes ranging from 24 in. to 32 in. Fifteen participants used the Chrome browser, 2 used Safari, while one used Firefox for the tasks.

The setup, tasks, and durations were decided based on a within-subjects pilot of the study described above with 2 participants: one native and one non-native English speaker. ConceptScope and DocuBurst employed different datasets in this study. The decisions to suggest time durations for the questions and to set up the final study as a between-subjects study were made based on the long duration of each session and on the participant's fatigue toward the end of each session.

7.4 Procedure

Participants first responded to an online pre-survey providing their demographic and background information. Once they had finished familiarizing themselves with the interface, the participants performed the tasks described in Sec. 7.2. Participants followed a concurrent think-aloud protocol while executing the tasks, with the moderator recording their verbalizations and their screen through a videoconferencing application. Finally, the participants were invited to finish a brief survey about the tool and share their feedback about their experience with the interface, both as open-ended responses and on the NASA TLX scale [20].

8 RESULTS AND DISCUSSION

8.1 General Behavior Patterns

We categorized participants into two groups based on how they attempted to gather the information they needed to answer the questions, rather than how they used the tools in general. One group comprised participants that mainly used the visualization, and the other, those that mainly used the raw text display. Seven of the 9 participants who used ConceptScope primarily used the main Bubble Treemap visualization to glean the required information, while the remaining 2 relied more on the raw text reading from the document. In DocuBurst, only 5 of the 9 participants used the main Sunburst diagram as their main source of information, while 4 chiefly relied on close-reading of the text.

Participants using ConceptScope used the main visualization more than participants using DocuBurst. This was partly due to the raw text reading experience offered by the two interfaces, and partly due to the ability of the visualizations and the knowledge base in conveying a relevant overview. In ConceptScope, documents were split into sentences and displayed in a relatively small vertical space (see Fig. 2c). Therefore, participants tended to read only a few sentences prior to and after the key sentence for a specific task instead of going through larger blocks of text. As participant *Pc*7 stated, *"because my resolution is small and my mouse is sensitive, so when I move it jumps between the text very easily (in transcript view). And this box (the tooltip showing the corresponding sentence) doesn't include the complete paragraph, so it's easy to get lost..."*. In contrast, DocuBurst showed text as paragraphs in a view that used more vertical space, such that users were able to read the sentences more easily. "*One thing I like this system is when I click some words, they divide it as paragraph rather than the entire document...help me read more specifically*", said participant *Pd*3.

When answering a given question, 7 of the 9 participants using ConceptScope searched or explored related information in the interface and summarized their findings. The remaining 2 mainly attempted to recall the answer from earlier explorations and then referred to the interface to confirm. For DocuBurst, this distribution was 5 participants chiefly exploring the interface, and 4 chiefly recalling the answer. Compared to ConceptScope, more participants using DocuBurst answered questions from memory, almost equal to the number of participants who explored the visualizations to find answers. Participant comments indicated that they felt they might spend too much time in locating the required information. For instance, when trying to find common concepts between two documents (task T2), participant *Pd*9 who used DocuBurst commented that "*it is really hard to see all of them (words in the sunburst diagram). And I really wanna expand one of those, but then I'm not sure if it will cover all the things that I wanna see.... It's hard to go back to where you came from*". Similar comments were also made by those participants using DocuBurst to first gather information before answering the question. In this study, we did not screen participants based on document familiarity as their responses were valuable to us regardless of their prior knowledge of the document. We had 1 participant in each group (*Pc*9 and *Pd*5) who had read the document for T1. *Pc*9 answered questions faster than the other participants, while *Pd*5 used close reading rather than Docuburst's Sunburst diagram to answer questions, saying, "*I don't know how to use this tool to help me read this paper.*"

8.2 Task-Level Observations

We further separate task-wise participant behavior based on how they achieved specific objectives within tasks. This behavior was not restricted to any one task; rather, it characterized how certain participants chose to access information across tasks.

Document Sensemaking: When exploring the full document (T1), participants across both interfaces attempted to use the visualization to quickly get a sense of what topics were addressed in the document. Ten of the 18 participants (6 using ConceptScope, 4 using DocuBurst) were able to quickly identify that the document was an "InfoVis paper". Certain participant behaviors were similar across both interfaces. Most of them explored the document using the main visualization first, and only later resorted to close reading of the text. Even after recognizing it as an academic paper, only 2 participants relied on the paper structure (e.g., abstract/introduction) to get a sense of the document.

However, DocuBurst users were more easily overwhelmed by the large number of words in the Sunburst diagram, many of which (they felt) were not closely related to the main theme of the document. Participant *Pd*3 observed, "*some words maybe appear really frequently, but it's actually not very important ... it's just because it's used very frequently by any document.*" Another participant (*Pd*2) found it difficult to organize the words into themes, saying "*it is a little bit hard to place the information together, because you don't know what the correlation is between (among) these things (i.e. the concepts provided)*"

Participants' perception of the document when using ConceptScope was largely influenced by the extent of overlap between the document text and the ontology. For instance, the concept "visualization" being well-defined in the ontology, was successfully identified by 8 out of 9 participants in T1. However, the concept "animation" was not as well-defined in CSO, as a result of which 5 out of 9 participants failed to determine that task T2 involved papers discussing animation. In comparison, 8 of the 9 using DocuBurst were able to successfully identify the animation theme.

Concept Sensemaking: When making sense of a concept (**R2, R3**), most of the participants chose to locate it in the main visualization first, and only then looked at the other views to answer relevant questions. To locate a specific concept in the visualization, participants' strategies varied based on the solutions available in the interface and their preference.

In ConceptScope, 5 of the 9 participants used the search feature, while the others preferred to visually search the concept in the interface, i.e. looking it up in the concept list or directly checking the Bubble Treemap. Since DocuBurst did not feature a search box available, all 9 participants set the concept to locate as the root word. However, eight of the 9 participants failed with this strategy and had to set alternatives of the original concept (e.g. a parent concept, a synonym, or a substring of the target concept) as root words. One unique strategy that at least 3 participants used to search in DocuBurst was to start from higher-level concepts and dive deeper towards their targets in the sunburst diagram. Once again, their success depended on their choice of parent concepts: they often lost their way as they could not retrace their steps. In comparison, participants found it more straightforward to locate concepts in ConceptScope.

While participants using either interface chiefly attempted to define a concept (**R1**) by referring to the context of its use (**R2**), their approach to identify the context was different across the interfaces. In ConceptScope, the concordance view was used the most, with all 9 participants using this view to identify the context at least once. This was followed by the close reading of the transcript (used by 7 participants), with the word cloud being used by 6 participants at least once. Although DocuBurst also provided a word cloud, only 2 participants used it for context. This was likely because DocuBurst's word cloud was not organized into concepts as done in ConceptScope, and furthermore, the word cloud in DocuBurst—designed to supplement the main visualization—only featured proper nouns that would not otherwise be visualized in the Sunburst diagram. To find related concepts (**R3**), participants using ConceptScope chiefly referred to the Bubble Treemap while DocuBurst users referred to the raw text view.

Multi-document Comparison: We observed participants' behavior when comparing documents both at the conceptual level and the full-text level (**R4**). Participants using ConceptScope used several techniques including highlighting concepts in the Bubble Treemap, highlighting concepts in the concept list, checking the relevant sparklines, and comparing the word cloud within a concept group. Five of the 9 participants reported that these techniques were sufficient to answer all of the questions in tasks T2 and T3. Participant *Pc*1 observed, "*just looking at this (the Bubble Treemap for the third document in T3), you can see some colors are different, means some different concepts exist here... you can immediately see it*". When the visual clues were not enough to aid them to summarize the similarities or differences between/among the documents, the other 4 participants resorted to close reading of the document.

In contrast, most of the participants using DocuBurst mentioned that the visualizations and interactions were not sufficient to help them compare the concepts or full text of the documents. Participant *Pd*5 commented, "*the visual encoding (distinguishing concepts between documents) is confusing to me*". Participant *Pd*9 felt "*it is really hard to see all of them (concepts)*" when they tried to identify unique concepts of one document. Both participant *Pd*1 and *Pd*8 were distracted by such general words as "*part*" and "*paper*", because they were the only few words marked as being shared by both documents. As a solution, they chose to read the document text closely to make sure their responses to the questions were accurate enough.

8.3 Overall Feedback

Fig. 5 shows the difference in participant experience for the study between ConceptScope and DocuBurst. We can see from the figure that participants' experience was more or less similar between the two interfaces with the exception of frustration: participants using ConceptScope were less frustrated ($Md = 2, IQR = 1$) than those using DocuBurst ($Md = 4, IQR = 4$). Observation and feedback indicated that participants using DocuBurst found themselves distracted by less relevant concepts. Participant *Pd*3 stated that the interface didn't provide "important" keywords as expected: "*When I click 'person'... it (the corresponding sector in sunburst diagram) is really big, means that it is important. However, I don't think it is important based on what I've seen*". Participant *Pd*4 mistook the document in task T1 for a medical paper and participant *Pd*9 mistook those in T2 as related to chemistry, based on their (mistaken) interpretation of proper nouns in the word cloud.

As general feedback, most participants using ConceptScope considered it suitable to provide an overview for unfamiliar documents, while those using DocuBurst felt it was better suited as a supplementary tool when exploring familiar documents. Typical comments about ConceptScope included "*these multiple views are nice and easy to understand*"(participant *Pc*2), "*it seems like a pretty useful tool especially for exploring large set of documents to get an idea of what the main topics are, what kind of researchers are active*" (*Pc*7). With DocuBurst, participant *Pd*2 suggested that "*the tool should be used as a supplementary tool ... doesn't help too much with understanding the document*". In addition, participants also reflected that the learning curves for both tools were relatively steep. "*It was hard at the beginning, but not so hard later*", commented participant *Pc*4.

When regarding the features of each interface, the Level Slicer (Sec. 5.2.1) in ConceptScope was marked as least useful by participants. Participant *Pc*1 observed that "*the level slicer is probably useful if the document is extremely complex ... but this dataset is relatively simple*". Three of the participants thought the list view (Sec. 5.1.2) was the most useful feature. We also observed 7 participants used it for comparison tasks and 4 participants used it to search target concepts in the study. Only 2 participants rated the Bubble Treemap (Sec. 5.1.1) as the most useful feature, while one marked it as the least useful one. Yet, we did see 7 participants used it as their major source of visual clues when comparing multiple documents. It is likely that the participants used the Bubble Treemap as providing supplementary information to the concept list view, which they found to be most useful.

9 LIMITATIONS AND FUTURE WORK

Based on participant behavior and feedback, we illustrate that ConceptScope's ontology-based visualization and grouped word clouds help participants define and contextualize concepts, and—for a given concept—explore other concepts related to it. On the other hand, ConceptScope's domain dependency makes it less suitable for reviewing text that spans multiple disciplines. In contrast, DocuBurst's domain-agnostic reference (i.e. WordNet) allows it to be applied more widely, though the overviews are less useful when highly domain-specific content is visualized. In addition, DocuBurst's interface is more amenable to close reading of the document.

Our study can be considered preliminary, as we were interested in participants' exploratory behavior, insights, and comprehension. We plan to conduct longitudinal studies to evaluate the utility of ConceptScope as a tool for preliminary review and further exploration before and after close-reading of documents, and examine additional encodings such as position constancy of concepts in the bubble treemap for document comparison.

In the future, we plan to address issues relating to the ontology lookup. One main disadvantage is the dependence on ontologies that may or may not be mature. We currently use DBPedia to "broaden" our lookup, but using DBPedia detracts from the strict definitions and relationship requirements to which domain ontologies need to adhere. Our Bubble Treemap visualization as well as our ontology lookup can currently support only one ontology. This makes it difficult to view documents of an interdisciplinary nature. We also intend to explore the application of our approach to real-time visualizations of online forums or technical communication in the form of emails or instant messengers.

10 CONCLUSION

In this paper, we proposed ConceptScope, an interface that aids a knowledge-based exploration and comparison of documents based on a reference domain ontology. We present the use of a Bubble Treemap visualization as the primary overview visualization to show the distribution of concepts for a document of interest, and describe our approach to translate document content into appropriate queries that best reflect the concept spread and show their hierarchical relationships in the domain ontology. We illustrate our approach using the Computer Science Ontology as our reference. We demonstrate the use of ConceptScope for document exploration and comparison, and then evaluate ConceptScope against DocuBurst, the only other overview visualization based on human-curated knowledge. We find that ConceptScope offers greater advantages in terms of domain-specificity, contextual views, and comparison of multiple documents, but not for close reading of documents, or documents spanning multiple domains. DocuBurst's domain-agnosticism makes it more suitable for a general-purpose document exploration tool spanning multiple domains, but less so for multi-document comparison or in-depth, domain-specific exploration. Our future research aims to address this issue by enabling the use

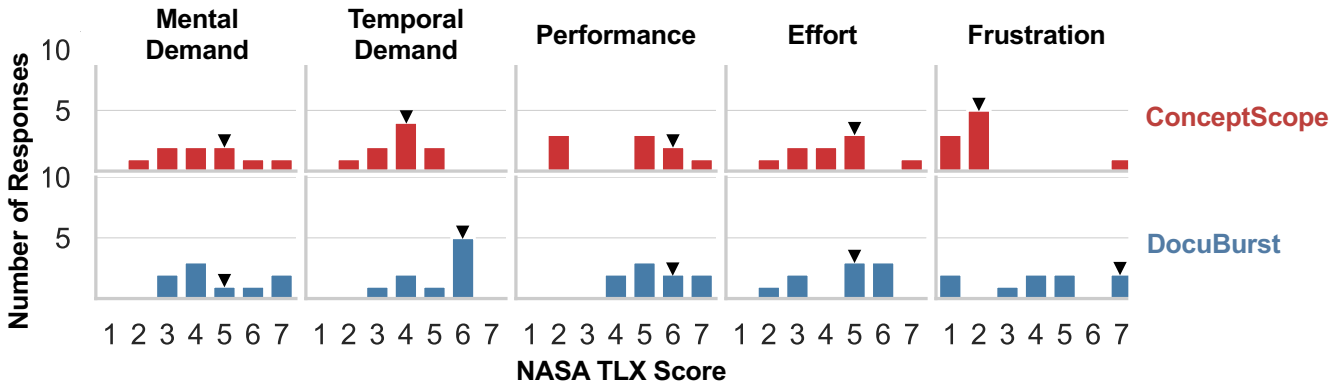


Figure 5: Distribution of NASA TLX responses showing participant feedback towards ConceptScope and DocuBurst. A ▼ symbol indicates where a user familiar with the document (*Pc*9 for ConceptScope and *Pd*5 for DocuBurst) has rated their experience.

of multiple reference ontologies, and explore text content such as online forums and organizational communication.

ACKNOWLEDGMENTS

We are grateful to Prof. Christopher Collins and Nathan Beals for their help with using DocuBurst for our user study, to Sandra Bae for her narration in our demo video, and to Oh-Hyun Kwon for his suggestions to improve the paper. We also thank the participants who volunteered with our study during these challenging times and the anonymous reviewers for their feedback and suggestions that helped improve the quality of this paper. This research is sponsored in part by the U.S. National Science Foundation through grant IIS-1741536.

REFERENCES

[1] Nassira Achich, Bassem Bouaziz, Alsayed Algergawy, and Faiez Gargouri. 2017. Ontology Visualization: An Overview. In *International Conference on Intelligent Systems Design and Applications*. Springer, Cham, USA, 880–891. https://doi.org/10.1007/978-3-319-76348-4_84

[2] Eric Alexander, Joe Kohlmann, Robin Valenza, Michael Witmore, and Michael Gleicher. 2014. Serendip: Topic model-driven visual exploration of text corpora. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*. IEEE, USA, 173–182. https://doi.org/10.1109/vast.2014.7042493

[3] Daniel Archambault and Helen C Purchase. 2016. Can animation support the visualisation of dynamic graphs? *Information Sciences* 330 (2016), 495–509. https://doi.org/10.1016/j.ins.2015.04.017

[4] Todd Barlow and Padraic Neville. 2001. A comparison of 2-D visualizations of hierarchies. In *IEEE Symposium on Information Visualization*. IEEE, USA, 131–138. https://doi.org/10.1109/infvis.2001.963290

[5] Benjamin B Bederson and Angela Boltman. 1999. Does animation help users build mental maps of spatial information?. In *Proceedings IEEE Symposium on Information Visualization*. IEEE, USA, 28–35. https://doi.org/10.1109/infvis.1999.801854

[6] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., USA. https://doi.org/10.1007/s10579-010-9124-x

[7] Nick Bostrom. 2015. What happens when our computer gets smarter than we are? https://www.ted.com/talks/nick_bostrom_what_happens_when_our_computers_get_smarter_than_we_are?language=en Accessed December 3, 2019.

[8] Christopher Collins, Sheelagh Carpendale, and Gerald Penn. 2009. DocuBurst: Visualizing Document Content using Language Structure. *Computer Graphics Forum* 28, 3 (2009), 1039–1046. https://doi.org/10.1111/j.1467-8659.2009.01439.x

[9] Christopher Collins, Fernanda B Viegas, and Martin Wattenberg. 2009. Parallel tag clouds to explore and analyze faceted text corpora. In *IEEE Symposium on Visual Analytics Science and Technology*. IEEE, Atlantic City, NJ, USA, 91–98. https://doi.org/10.1109/VAST.2009.5333443

[10] Weiwei Cui, Xiaoyu Zhang, Yun Wang, He Huang, Bei Chen, Lei Fang, Haidong Zhang, Jian-Guan Lou, and Dongmei Zhang. 2019. Text-to-Viz: Automatic Generation of Infographics from Proportion-Related Natural Language Statements. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2019), 906–916. https://doi.org/10.1109/TVCG.2019.2934785

[11] Marek Dudáš, Steffen Lohmann, Vojtěch Svátek, and Dmitry Pavlov. 2018. Ontology visualization methods and tools: a survey of the state of the art. *The Knowledge Engineering Review* 33 (2018), e10. https://doi.org/10.1017/S0269888918000073

[12] Mennatallah El-Assady, Valentin Gold, Carmela Acevedo, Christopher Collins, and Daniel Keim. 2016. ConToVi: Multi-party conversation exploration using topic-space views. *Computer Graphics Forum* 35, 3 (2016), 431–440. https://doi.org/10.1111/cgf.12919

[13] Mennatallah El-Assady, Rita Sevastjanova, Bela Gipp, Daniel Keim, and Christopher Collins. 2017. NEREx: Named-Entity Relationship Exploration in Multi-Party Conversations. *Computer Graphics Forum* 36, 3 (2017), 213–225. https://doi.org/10.1111/cgf.13181

[14] Christiane Fellbaum (Ed.). 1998. *WordNet: An electronic lexical database*. MIT press, USA.

[15] Samah Gad, Waqas Javed, Sohaib Ghani, Niklas Elmqvist, Tom Ewing, Keith N Hampton, and Naren Ramakrishnan. 2015. ThemeDelta: dynamic segmentations over temporal topic models. *IEEE Transactions on Visualization and Computer Graphics* 21, 5 (2015), 672–685. https://doi.org/10.1109/TVCG.2014.2388208

[16] Michael Glueck, Peter Hamilton, Fanny Chevalier, Simon Breslav, Azam Khan, Daniel Wigdor, and Michael Brudno. 2015. PhenoBlocks: phenotype comparison visualizations. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2015), 101–110. https://doi.org/10.1109/TVCG.2015.2467733

[17] Jochen Görtler, Christoph Schulz, Daniel Weiskopf, and Oliver Deussen. 2017. Bubble treemaps for uncertainty visualization. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 719–728. https://doi.org/10.1109/TVCG.2017.2743959

[18] Brynjar Gretarsson, John O'donovan, Svetlin Bostandjiev, Tobias Höllerer, Arthur Asuncion, David Newman, and Padhraic Smyth. 2012. TopicNets: Visual analysis of large text corpora with topic modeling. *ACM Transactions on Intelligent Systems and Technology* 3, 2 (2012), 23. https://doi.org/10.1145/2089094.2089099

[19] Thomas R. Gruber. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5, 2 (1993), 199–220. https://doi.org/10.1006/knac.1993.1008

[20] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, USA, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[21] Sebastian Hellmann. 2015. DBpedia lookup | DBpedia. https://wiki.dbpedia.org/lookup Accessed October 3, 2019.

[22] Nathalie Henry, Jean-Daniel Fekete, and Michael J McGuffin. 2007. NodeTrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics* 13, 6 (2007), 1302–1309. https://doi.org/10.1109/TVCG.2007.70582

[23] Jeremy Howard. 2014. The wonderful and terrifying implications of computers that can learn. https://www.ted.com/talks/jeremy_howard_the_wonderful_and_terrifying_implications_of_computers_that_can_learn?language=en Accessed December 3, 2019.

[24] Stefan Jänicke, Greta Franzini, Muhammad Faisal Cheema, and Gerik Scheuermann. 2015. On Close and Distant Reading in Digital Humanities: A Survey and Future Challenges.. In *EuroVis (STARs)*. The Eurographics Association, 83–103. https://doi.org/10.2312/eurovisstar.20151113

[25] Akrivi Katifori, Constantin Halatsis, George Lepouras, Costas Vassilakis, and Eugenia Giannopoulou. 2007. Ontology visualization methods—a survey. *Comput. Surveys* 39, 4 (2007), 10. https://doi.org/10.1145/1287620.1287621

[26] Daniel A Keim and Daniela Oelke. 2007. Literature fingerprinting: A new method for visual literary analysis. In *IEEE Symposium on Visual Analytics Science and Technology*. IEEE, USA, 115–122. https://doi.org/10.1109/VAST.2007.4389004

[27] Peter Kovesi. 2015. Good colour maps: How to design them. *arXiv preprint arXiv:1509.03700* (2015).

[28] Kostiantyn Kucher and Andreas Kerren. 2015. Text visualization techniques: Taxonomy, visual survey, and community insights. In *IEEE Pacific Visualization Symposium*. IEEE, Hangzhou, China, 117–121. https://doi.org/10.1109/pacificvis.2015.7156366

[29] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195. https://doi.org/10.3233/SW-140134

[30] Christopher "moot" Poole. 2010. The case for anonymity online. https://www.ted.com/talks/christopher_moot_poole_the_case_for_anonymity_online Accessed December 3, 2019.

[31] Franco Moretti. 2005. *Graphs, maps, trees: abstract models for a literary history*. Verso, USA.

[32] Tamara Munzner. 2009. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics* 15, 6 (2009), 921–928. https://doi.org/10.1109/TVCG.2009.111

[33] Jaume Nualart and Mario Pérez-Montoro. 2013. Texty, a visualization tool to aid selection of texts from search outputs. *Information Research* 18, 2 (2013).

[34] Daniela Oelke, David Spretke, Andreas Stoffel, and Daniel A Keim. 2011. Visual readability analysis: How to make your writings easier to read. *IEEE Transactions on Visualization and Computer Graphics* 18, 5 (2011), 662–674. https://doi.org/10.1109/TVCG.2011.266

[35] Daniela Oelke, Hendrik Strobelt, Christian Rohrdantz, Iryna Gurevych, and Oliver Deussen. 2014. Comparative exploration of document collections: a visual analytics approach, In Computer Graphics Forum. *Computer Graphics Forum* 33, 3, 201–210. https://doi.org/10.1111/cgf.12376

[36] Byeong-Min Roh, Soundar RT Kumara, Timothy W Simpson, and P Witherell. 2016. Ontology-Based Laser and Thermal Metamodels for Metal-Based Additive Manufacturing. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers (ASME), United States, 21–24. https://doi.org/10.1115/DETC2016-60233

[37] Angelo A Salatino, Thiviyan Thanapalasingam, Andrea Mannocci, Francesco Osborne, and Enrico Motta. 2018. The computer science ontology: a large-scale taxonomy of research areas. In *International Semantic Web Conference*. Springer International Publishing, Cham, 187–205. https://doi.org/10.1007/978-3-030-00668-6_12

[38] Ben Shneiderman. 1992. Tree Visualization with Tree-Maps: 2-d Space-Filling Approach. *ACM Transactions on Graphics* 11, 1 (1992), 92–99. https://doi.org/10.1145/102377.115768

[39] Ben Shneiderman. 2003. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*. Morgan Kaufmann, San Francisco, 364–371. https://doi.org/10.1016/B978-155860915-0/50046-9

[40] John Stasko, Richard Catrambone, Mark Guzdial, and Kevin McDonald. 2000. An evaluation of space-filling information visualizations for depicting hierarchical structures. *International journal of human-computer studies* 53, 5 (2000), 663–694. https://doi.org/10.1006/ijhc.2000.0420

[41] John Stasko, Carsten Görg, and Zhicheng Liu. 2008. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization* 7, 2 (2008), 118–132. https://doi.org/10.1057/palgrave.ivs.9500180

[42] Margaret-Anne Storey, Mark Musen, John Silva, Casey Best, Neil Ernst, Ray Fergerson, and Natasha Noy. 2001. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. In *Workshop on interactive tools for knowledge capture*, Vol. 73.

[43] Zeynep Tufekci. 2016. Machine intelligence makes human morals more important. https://www.ted.com/talks/zeynep_tufekci_machine_intelligence_makes_human_morals_more_important/transcript?referrer=playlist-talks_on_artificial_intelligen#t-3550 Accessed October, 2020.

[44] Frank Van Ham, Martin Wattenberg, and Fernanda B Viégas. 2009. Mapping text with phrase nets. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1169–1176. https://doi.org/10.1109/TVCG.2009.165

[45] Fernanda B Viégas, Martin Wattenberg, and Jonathan Feinberg. 2009. Participatory visualization with wordle. *IEEE transactions on visualization and computer graphics* 15, 6 (2009), 1137–1144. https://doi.org/10.1109/TVCG.2009.171

[46] w3.org. 2013. SPARQL 1.1 Query Language. https://www.w3.org/TR/sparql11-query/ Accessed June 9, 2019.

[47] Weixin Wang, Hui Wang, Guozhong Dai, and Hongan Wang. 2006. Visualization of large hierarchical data by circle packing. In *Proceedings of the ACM CHI conference on Human Factors in computing systems*. Association for Computing Machinery, New York, NY, United States, 517–520. https://doi.org/10.1145/1124772.1124851

[48] Martin Wattenberg and Fernanda B. Viégas. 2008. The Word Tree, an Interactive Visual Concordance. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1221–1228. https://doi.org/10.1109/TVCG.2008.172

[49] Furu Wei, Shixia Liu, Yangqiu Song, Shimei Pan, Michelle X Zhou, Weihong Qian, Lei Shi, Li Tan, and Qiang Zhang. 2010. Tiara: a visual exploratory text analytic system. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, New York, NY, United States, 153–162. https://doi.org/10.1145/1835804.1835827

[50] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. 2007. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1129–1136. https://doi.org/10.1109/TVCG.2007.70589

[51] Paul Witherell, Sundar Krishnamurty, and Ian R Grosse. 2007. Ontologies for supporting engineering design optimization. *Journal of Computing and Information Science in Engineering* 7, 2 (2007), 141–150. https://doi.org/10.1115/1.2720882