CLIOQUERY: Interactive Query-Oriented Text Analytics for Comprehensive Investigation of Historical News Archives

ANONYMOUS AUTHOR(S)

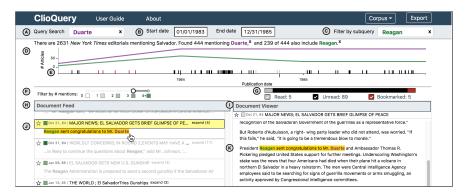


Fig. 1. CLIOQUERY, an interactive text analytics system for helping historians investigate queries in news archives. Features (letters A to K) include: (D) a Time Series View showing the frequency of a user's query through time, (H) a linked Document Feed showing a skimmable query-oriented summary of every mention of the query in the corpus, and (I) a linked Document Viewer showing a selected news story, with text from the query-oriented summary highlighted in yellow. Section 4 describes the full system, explaining each feature.

Historians and archivists often find and analyze occurrences of query words in newspaper archives, to help answer fundamental questions about society. But much work in text analytics focuses on helping people analyze other textual units, such as events, clusters, ranked documents, entity relationships or thematic hierarchies. We thus propose CLIOQUERY, a text analytics system uniquely organized around the analysis of query words in context. To help historians investigate query words, CLIOQUERY tests new text simplification techniques from natural language processing in a visual analytics system, by pairing such NLP methods with more traditional features to help social researchers quickly and comprehensively gather and analyze all occurrences of a query word across an archive, with minimal algorithmic bias. The tool's design and feature set emerged from an investigation into the needs of historians and archivists. In two separate user studies, historians explained how CLIOQUERY's novel summarization features can help facilitate historical research; [Abe: a separate quantitative comparison study also suggests advantages over a keyword document search baseline.] Such results suggest possible new directions for text analytics in other query-oriented settings.

CCS Concepts: • Human-centered computing \rightarrow Interactive systems and tools.

Additional Key Words and Phrases: interactive text analytics, digital humanities, user interfaces

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

^{© 2021} Association for Computing Machinery.

^{2160-6455/2021/1-}ARTACM Article No \$15.00

https://doi.org/xxx

ACM Article No:2 Anon.

ACM Reference Format:

Anonymous Author(s). 2021. CLIOQUERY: Interactive Query-Oriented Text Analytics for Comprehensive Investigation of Historical News Archives. *ACM Trans. Interact. Intell. Syst.* xx, xx, Article ACM Article No (January 2021), 39 pages. https://doi.org/xxx

1 INTRODUCTION

Newspaper archives are fundamental resources for historians, librarians and social scientists [2, 17], because they offer a detailed primary source record of how social processes evolve across time [103]. For instance, social researchers have used news archives to examine vital questions such as why the United States abolished slavery [45] and how different jurisdictions slowed the spread of the 1918 flu [83]. While historians are known to use archives in different ways (e.g. sequential browsing [2]), prior work reports that historians often look for "specific keywords" [2, p. 2] in newspaper corpora. For instance, scholars in history and social science journals describe tracking down and reviewing occurrences of words like "William Benbow" [106], "Frances Maule" [121], "watermelon" [8], "Japanese beetles" [117], "refugee" [102], "Loving" [62], and "race suicide" [79] in news archives to help answer questions about society. In this work, we describe such search terms (e.g. "William Benbow") as **queries** and we describe each exact occurrence of a query in an archive as a query **mention**. (Section 10 discusses possible improvements to exact string matching.) We then describe the task of locating query mentions as **mention gathering**, and the closely-related task of reviewing and drawing conclusions from query mentions within the context of surrounding text as **mention analysis**. We formally define these terms and tasks in Section 3.1.

Much prior work in interactive text analytics (Section 2) does not focus on helping people investigate mentions of a query word in context. Instead, prior systems (designed for different use cases) focus on the analysis of other latent and observable textual units, such as topics [72], events [76], document metadata [38], clusters [11], interrelated entities [49] or thematic hierarchies [12]. Using terminology from Heer *et al.* [21], who articulate best practices for text analytics, because such systems do not use query words in context as their central "unit of analysis," they offer "visual encodings", "modeling decisions" and interactions which are are poorly "alligned" to the "tasks, expectations, and background knowledge" of historians and archivists.

This misalignment means that prior text analytics systems have concrete downsides for mention gathering and analysis. For example, some prior systems focused on helping people analyze high-level text units such as temporal trends in word use (e.g. ThemeRiver [56]) do not show query words in underlying text. Similarly, other systems offer only indirect and incomplete access to query words in context, via extraneous mediating abstractions. We detail these limitations in Section 2.3.

Additionally, in this work, we also study the current practices of social researchers such as Shinozuka [117] and others [8, 62, 79, 102, 106, 121], who report using traditional keyword document search engines like ProQuest [105] to gather and analyze query mentions in newspaper archives and other corpora. (We use *corpus* and *archive* interchangeably; we assume the corpus is an archive.) Traditional keyword document search tools return relevance-ranked document lists in response to a free-text query. We propose they are baselines for mention gathering and analysis (Section 2.2.1), because they are widely used in historical practice [2, 17, 106, 121].

Yet keyword document search tools also have limitations for finding and analyzing query words in context. First, because almost all words are very rare (a well-known property of text [139]), any given query will very likely appear only a small number of times within a document. This means that people reviewing a ranked document list will have to examine many passages within documents that do not directly mention their query term. While search within document features

¹Using terminology from prior work [104], it is possible to interpret mention gathering as a kind of information foraging and mention analysis as a kind of sensemaking.

User study	Num. participants	Total hours
Needfinding study to guide system design (Sec. 3)	5	4.5
Expert interview study to evaluate CLIOQUERY features (Sec. 5)	5	5
Field study to test CLIOQUERY in the wild (Sec. 7)	2	5

Table 1. This work presents three separate user studies with historians and archivists. Section 3 describes institutional approval. Tables in the Appendix describe the backgrounds of participants in greater detail. [Abe: TODO quant study add. total hours=1/3]

(e.g. control + F in Chrome [24]) can certainly help, gathering and analyzing query words in context using a keyword document search system still requires opening each article in its own window or tab, locating mentions within the article, reading passages which mention the query, and integrating information from such passages with existing knowledge, before moving on to the next document in the corpus. This means that people must context switch across stories as they perform a multi-step process to gather and analyze query mentions, keeping track of information from one document as they jump to the next (see Figure 2). Navigating between documents is thought to impose cognitive costs in keyword document search tools [41, 138], and context switching across views is thought to impose cognitive costs in visual analytics systems [135]. [Abe: a forward ref to quant study would be good here]

Noting the importance of historical investigation and the limitations of existing tools (for mention gathering and analysis), we propose the CLIOQUERY³ text analytics system, to help historians in their work investigating query words in an archive. Unlike prior tools, which focus on the analysis of other textual units like topics or hierarchies (Section 2.3), CLIOQUERY is designed and built to help people analyze query words in context, reflecting the needs and practices of historians and archivists. Creating tools around the "tasks, expectations and background knowledge" of end users is believed to be a best practice in text analytics [21]; we thus studied and solicited feedback from such experts in designing and prototyping [46] CLIOQUERY.

In particular, working closely with historians and archivists revealed intertwined technical and design requirements (Section 3), which guided our work. First, because our needfinding investigations revealed the importance of comprehensive review in historical research (Section 3.3.2), CLIOQUERY includes a novel Document Feed feature that shows a query-focused summary of every single mention of a user's query term across a corpus. To create such summaries, CLIOQUERY applies new text simplification techniques from natural language processing (NLP), and then tests these methods in a user-facing setting, by combining such techniques with more traditional features from visual analytics. For instance, because we found that historians require transparency and contextual information to interpret evidence, CLIOQUERY's novel Document Feed is presented alongside a more traditional linked full-text Document View, designed to quickly and transparently show text from the summary within the context of full-length documents. Similarly, because analytics tools often show an "overview first" [118] and because temporal analysis is crucial to historians, CLIOQUERY also includes an interactive Time Series View to provide an overview of a query through time. Together, through these and other features (Section 4), CLIOQUERY offers a query-oriented approach to text analytics, which allows historians to facilitate mention gathering and analysis.

In total, our work offers the following:

²Showing a single document in a single window or tab is a common interface pattern. It is employed, for instance, in the Overview [12] and Jigsaw [122] document viewers, and in traditional search user interfaces, which often link to individual documents from a main search engine results page [28, Section 6.3].

³Clio is a common prefix (e.g. ClioVis [14]), implying a connection with history.

ACM Article No:4 Anon.

• A synthesis of extensive prior research in text analytics (Section 2). In reviewing prior work on interactive analysis of text across time, we found that many efforts from the NLP, HCI and Visualization communities focus on offering overviews of corpus contents. Such overviews might help users formulate queries, but are not designed for mention gathering and analysis, when the user already knows what to search for.

- An investigation into user needs and requirements (Section 3). To build our tool, we translated prior research on historians' information-seeking behavior into concrete guidelines for system design. We also validated and contextualized prior work by conducting five needfinding interviews with historians and archivists, and by gathering feedback on early prototypes. This process revealed a need for transparency, trustworthiness, context and comprehensiveness in archival tools, which might inform future work on historical search [119, 120] and text summarization [32, 93].
- The CLIOQUERY system (Section 4). CLIOQUERY is an open-source (URL withheld), text analytics system designed for historians. Unlike much prior work in text analytics, which helps people analyze textual units like topics, events or thematic hierarchies (Section 2.3), CLIOQUERY is designed for the analysis of query mentions in context. CLIOQUERY combines novel query-focused summarization methods with more traditional features like linked views, in-text highlighting and time series plots to help experts quickly, comprehensively and transparently find and review query mentions across an archive.
- An evaluation of specific CLIOQUERY features (Sections 5 and 6). To test the utility and usability of CLIOQUERY, we conducted an expert interview study with five social researchers, who used the system to answer a historical question from news archives. After methodically coding qualitative feedback, we learned that many experts found CLIOQUERY's skimmable summaries useful, because they condensed documents to facilitate quick review of query mentions. We also learned that linking summary text with underlying source documents using in-text highlighting was essential, because it offered necessary context for interpreting summary output.
- An evaluation of CLIOQUERY in the wild (Section 7). To test CLIOQUERY in a realistic setting, we deployed the system in a field study with two historians, who used CLIOQUERY to answer questions from their own research. In comparing experiences with CLIOQUERY to prior experiences with keyword document search systems, one historian explained how CLIOQUERY reduced their reading burden, and another explained how text summarization features facilitated rapid mention gathering and analysis. [Abe:
- A quantitative comparison with keyword document search tools (Section 8). We compare CLIOQUERY with a baseline keyword document search search system in a crowdsourced task, modeled on a real historical research question. We observe that participants who use CLIOQUERY features to complete the task correctly answer more historical reading comprehension questions than participants who complete the task using the baseline tool.]

We conclude by discussing our findings (Section 9), reviewing limitations and future work (Section 10), and describing possible applications of features and ideas from CLIOQUERY in other query-oriented settings, beyond historical research (Section 11).

2 RELATED WORK

 Historians sometimes gather and analyze mentions of specific query words in archives (Section 1). However, much prior work from the HCI, Visualization and NLP communities focuses on helping people gain high-level overviews of large bodies of text. We review this overview-oriented literature in Section 2.1. In Section 2.2, we also review another literature on search-based systems, which

CONTROL TO CREW:
GOOD AND BAD NEWS

GOOD AND BAD NEWS

GOOD AND BAD NEWS

**Sality Ride"

**Sality Ride"

**Sality Ride"

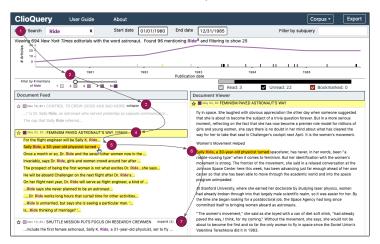
**Sality Ride"

**Sality Ride"

**Sality Ride **

**Sality Rid

(a) A user investigates Sally Ride by performing mention gathering and analysis using the *New York Times* web archive [95], a baseline keyword document search interface (Section 2.2.1). They first (1) click the top headline on the search engine results page (left) in order to (2) open a document in a new tab (shown on the right) and then (3) scroll down to locate mentions of "Ride" in the linked news story. The user reads and analyzes these mentions and then (4) context switches to the second document by clicking the second headline on the results page. This (5) opens a new story in a new tab. For this second document, they (6) use a search in document feature [24] (i.e. Control+F) to help locate mentions of "Ride" within the story.



(b) A user (1) searches for "Ride" using CLIOQUERY and (2) sets the filter-by-count slider to limit results to stories with at least two mentions of "Ride". The user then clicks the expand/collapse button on two news stories (3 and 4) to review all mentions of Ride from each story in the Document Feed. They then (5) click one shortened sentence mentioning "Ride" (6) to read it within the context of the full original document in the linked Document Viewer, with help from automatic in-text highlighting. The user then prepares to (7) click expand to review additional mentions of Ride in the next story.

Fig. 2. Reviewing mentions of U.S. astronaut Sally Ride in *The New York Times*, using CLIOQUERY (bottom) and a keyword document search tool (top). This particular example comes from our field study (Section 7), where one historian commented on the advantages of the CLIOQUERY interface over a baseline keyword document search system. "What can I do here [with CLIOQUERY] that I can't do there [with New York Times search]?" she said. "It's exploring this left-hand Document Feed." Where CLIOQUERY facilitates quick and comprehensive review of all query mentions, the keyword document search tool requires the user to read unnecessary passages and context switch across documents.

ACM Article No:6

focus on retrieving text from a corpus in response to a user query. This search-based approach seems better suited to mention gathering and analysis, as search-based systems can help historians find and review query mentions in a corpus. Much evidence (Section 1 and Table 2) also suggests that search-based systems are central to contemporary historical practice.

In presenting prior work, we emphasize common user interface design patterns [128], shared among multiple prior systems. Interface design patterns are "concrete bundles of components" [129] that help a user achieve some task. We say that *overview design patterns* help people survey the contents of archives, and that *search design patterns* help people query for specific selections from a body of text. Table 2 offers a summary of major design patterns from prior work. Some individual systems (e.g. Expedition [119]) may implement both overview and search patterns.

2.1 Overview design patterns

 2.1.1 Word and document clustering. Because people often can not review every document in a large corpus, many prior text analytics tools such as Termite [20], TIARA [72], Overview [12], RoseRiver [30], TextFlow [29], Serendip [1] and ConVisIT [63] try to suggest overall themes in a body of text, by identifying and displaying groups of thematically-related words and documents in a user interface. We describe this approach as the word and document clustering design pattern.

Some systems which implement the word and document clustering pattern are based on prior work from NLP, information retrieval, and text mining, focused on identifying and representing patterns of co-occurring words using methods such as topic models [9] and word embeddings [90].⁴ Researchers in HCI and Visualization extend this work by considering how to present such patterns in a graphical interface; some systems show changes in cluster patterns across time [29, 30, 72] (e.g. Figure 3a), others do not show time-based topics [12, 20]. Because automatic clusters may not match human mental models of a corpus, one line of work investigates human-in-the-loop techniques, which allow people to modify word or document clusters through interactions with a GUI [11, 63, 64, 67, 68, 99, 116].

Word clustering has a clear role in historical research. In query-oriented settings, clustering methods may help people formulate queries they had not considered [131]. Moreover, specialized and computationally-oriented digital humanists [92] and historians [51] have used word clusters from topic models for historical literary analysis. Nevertheless, successful application of topic modeling requires specialized knowledge and extensive interpretive effort [5, 111], making this method less accessible to a broader audience of historians. Additionally, many historians approach archives looking for mentions of what Allen and Sieczkiewicz describe as "specific keywords" [2] rather than looking to explore word cluster overviews from a topic model API. Because we design for historians investigating known query terms (Section 1), we do not employ the word clustering pattern in the ClioQuery interface.

2.1.2 **Textual and visual summaries**. Rather than showing lists of related words to offer a corpus overview, a large body of work on text summarization from NLP [32] instead attempts to create short paragraphs which convey the most "important" information in a corpus, by selecting a collection of sentences or sentence fragments from input documents to form an output summary. (This is sometimes described as extractive summarization [32], because the output text is extracted from input text.) User-facing systems such as Newsblaster [88] and NSTM [3] apply this research by showing such textual summaries in a graphical interface. We say that such tools implement the

⁴Themail [133] clusters words by time (instead of by co-occurrence statistics), and EventRiver [76] clusters documents by both time and semantic content. Because these systems show related words and documents (i.e. related by time period), we say they implement the clustering pattern. Similarly, VisGets shows clusters (of document tags) defined by a user's selection in the interface [37], which we consider to be a form of clustering.

textual summary design pattern (Figure 3b). Other closely related work from text visualization considers how to present summary text in specialized visual layouts such as Document Cards [125], Phrase Nets [132], Actor Views [78], Word Trees [136], concordance graphs [77] and subject-verb-object browsers [123]. We say that these interfaces offer structured summaries, as they place summary text within some structured visual format (e.g. a directed graph [132]).

[Abe: need to draw the contrast]

 Like word clusters, both traditional text summaries and structured visual summaries do not seem to help with mention gathering and analysis. A user can't turn to these forms of summaries to find and review query mentions, because "important" sentences selected for inclusion in summary output may or may not contain a given query word. Moreover, traditional approaches typically do not explain *how* "important" information is chosen, which may be important in the history domain (Section 9.3).

However, two ideas from the text summarization literature may help historians perform mention gathering and analysis. First, work in query-focused summarization tries to identify the most salient information in a corpus, based on a user's query [93]. Historians might use such query-focused summaries to review keywords in text. Query-focused summaries which define *all* query mentions as important enough to warrant inclusion in summary output may be especially helpful (see Section 3.3.2). Second, work on sentence compression [42, 44, 70] tries to shorten individual sentences by removing words, usually for the purpose of including more (shortened) sentences in a fixed-length summary. These methods, or closely-related sentence fusion techniques [4], might be used to shorten passages containing query terms, to help people quickly review many mentions of a query in context. We apply these two ideas from text summarization in CLIOQUERY (see Section 4.3 and 4.7).

2.1.3 **Time series plot**. Instead of showing text to summarize corpus contents, time series plots present the frequency of documents or query-mentions across time to offer a visual (rather than textual) corpus overview. This pattern is often implemented in text analysis tools [33, 34, 82, 108] and keyword search systems [84, 96, 119]. We divide time series visualizations into two groups. One group [84, 89, 120] shows the frequency of a single query term across time (e.g. Figure 3c), often using a line chart. Another group shows the frequency of multiple terms (e.g. highest-count words) using a stacked area chart [6, 56], and may not require a user-supplied query. While time series plots alone can not be used for mention gathering and analysis because they do not show underlying text from a corpus, such visualizations can hint at important events or changes across documents (e.g. Michel *et al.* [89]). We implement this design pattern in CLIOQUERY (Section 4.2).

2.2 Search design patterns

2.2.1 Keyword document search (baseline). Traditional keyword document search tools return relevance-ranked lists of documents on a search engine results page (SERP), in response to a free-text query [80]. Because historians often use such tools in practice (Section 1), we consider these systems to be baselines for mention gathering and analysis.⁵

Although keyword document search tools are widely used (Table 2), these systems have clear downsides for finding and reviewing query mentions. First, keyword document search systems impose unnecessary burdens from reading and context switching. This is described in detail in Section 1, and quantified in Section 8. Additionally, keyword document search systems rank documents according to a computational model of relevance, which may be undesirable for historians as relevance-ranking introduces opaque algorithmic influence over qualitative conclusions (by

⁵One strand of humanities scholarship critically investigates how widespread adoption of keyword document search tools might be distorting traditional humanistic research [106, 121, 131].

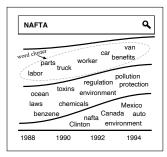
ACM Article No:8

Overview patterns (Section 2.1)

344

345

346



Summary of 129 documents

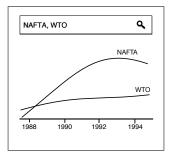
- Senior trade officials from the United States, Canada and Mexico failed here this

 weekend to reach agreements.
- weekend to reach agreements
 The North American Free Trade Agreement
- drew a bitter reaction tonight from unions

 Nafta that caused them to lose their jobs

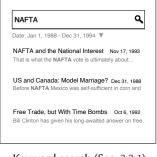
Sources

- 'Progress' but No Accord In 3-Nation ...
- Reaction; After Vote, Labor Is Bitter
- Sour Taste of Nafta: Old Friends Become ...
- (a) Word & doc. clustering (Sec. 2.1.1) Examples: [12, 20, 30, 63, 99, 116]
- (b) Textual & visual summary (Sec. 2.1.2) Examples: [3, 87, 88]



(c) Time series plot (Sec. 2.1.3) Examples: [6, 56, 89, 108]

Search patterns (Sec. 2.2)



NAFTA

That is what the NAFTA vote is ultimately ab
As a trade pact, NAFTA helps the economy
prices will fall But NAFTA's most important pri
ongress do with NAFTA as it pleased. Unles
matter of politics NAFTA was too important to
ho criticized pro-NAFTA votes. Tonight will te

(d) Keyword search (Sec. 2.2.1) Examples: [26, 57, 61, 95, 96, 108]

(e) Multi-doc. snippet (Sec. 2.2.2) Examples: [74, 100, 108, 133]

Fig. 3. We define five major user interface design patterns from prior work devoted to helping users understand news archives and other corpora (Section 2). Three of the design patterns focus on helping users gain an overview of archive contents (top two rows). Two of the design patterns focus on helping the user to search for specific documents or passages from a corpus (bottom row). Following Tidwell [128], this figure presents prototypical wireframes of each design pattern, created by the authors of this work. Each example above shows a system presenting results from 129 documents matching the query "NAFTA" on a corpus of *New York Times* editorials published between 1988 and 1994.

	Relevance ranking	Filter by Date	Known users
Chronicling America	✓	✓	I3, I4, I5,P4, H1
Newspapers.com	\checkmark	✓	I1,P1, P4, H2
New York Times Search	\checkmark	✓	I3, I4, I5, P1, P5, H1, H2
ProQuest	\checkmark	\checkmark	I1 to I5, P1, P3, P4, P5, H1, H2

Table 2. Example baseline keyword document search systems, featuring relevance-ranked search engine results pages and filtering by date; such features are common in many news archive interfaces [39]. Tables in the Appendix provide more details on the backgrounds of known users. Above, we use I1 to I5 to indicate all interviewees.

guiding people towards particular documents). Section 3 describes the importance of neutral and comprehensive review in historical research.

Ranking aside, keyword document search tools may also shape user perceptions of the contents of the individual documents in an archive, through displaying single-document summaries (also called query-biased snippets [130]) on the search engine results page. For example, Figure 3d displays three sample single-document summaries, showing what a computer deems to be the most important information from three different search results. Such single-document summaries may be inappropriate for historical research, as some historians may be skeptical of opaque models which select "important" information for their review (search engines try to include keywords in snippets, but do not try to explain summaries [28, Section 6.3.1]). Prototypes shown in the Appendix describe our own experiences attempting to apply similar document summarization techniques for historians without success.

2.2.2 Multi-document snippet. Where keyword document search systems return links to single documents in response to a user query, other systems return collections of smaller units like paragraphs, sentences or character spans, which are often drawn from multiple documents (see Figure 3e). We observe two different implementations of this multi-document snippet design pattern in interactive text analytics.

First, multi-document snippet features can be used in word clustering systems to help people investigate mentions of particular clustered words in context. For example, TIARA [72] allows analysts to review individual words from a cluster in underlying text. However, because TIARA is designed for showing broad themes rather than for reviewing query mentions, it does not comprehensively show all mentions of a given word in its multi-document snippet. Instead, TIARA chooses some selection of mentions for display, optimizing for diversity [72, Section 6]. Such curation may introduce unwanted algorithmic bias (Section 3.3.4), because the system chooses some but not all query mentions for display.

Additionally, other text analysis systems which are not necessarily focused on clustering sometimes include keyword-in-context (KWIC) views [100, 108, 134], showing each mention of a query word (or a selection of such mentions) on its own line of text, amid immediately surrounding tokens or characters (e.g. Figure 3e). While this form of multi-document snippet can be used for mention gathering and analysis, KWIC views have some limitations for historical research. First, in many cases, historians need to investigate particular query mentions within the context of full documents (Section 3.3.3). While KWIC views may include links to underlying sources, jumping from KWIC views to documents requires context switching into new windows or tabs to gather

⁶Google sometimes shows complex results snippets on the SERP, using proprietary techniques. Brin and Page briefly mention the need for such "Result Summarization" in their original paper [13, Section 6.1].

ACM Article No:10 Anon.

442

443

445

446

447

449

451

452

453 454

455

456

457

458 459

463

467

469

471

473

474

475

476

477

478

479

480

481

482

483

484 485

486

487

488

489 490 and analyze evidence. We explain why this is undesirable in Section 1. Second, KWIC views always show some number of pixels, characters or words immediately surrounding each query mention. This may result in awkward-sounding or choppy snippets that do not include the most salient information in source sentences; evidence suggests that people dislike awkward-sounding snippets [22]. Finally, KWIC views do not offer a way to keep track of which mentions have been reviewed during analysis, which may be important in historical research (Sections 3.3.2 and 4.5). Noting these shortcomings, it is possible to interpret certain CLIOQUERY features (Section 4.3 and 4.4) as a particular form of KWIC view, addressing some of these limitations. Situating CLIOQUERY withing the broader literature on visual analytics for text

Researchers have proposed many approaches to interactive text analytics [1, 11, 12, 20, 21, 30, 38, 49, 63, 72, 76, 99, 116]. Within this broad literature, CLIOQUERY is unique because it is designed to help people find and analyze all occurrences of a query word across a corpus (see Section 3.1). Using terminology from Heer et al., CLIOQUERY differs from prior work because its central "unit of analysis" [21] is the query word in context; the systems's "visual encodings", "modeling decisions" [21] and user interactions were all designed to help people quickly and comprehensively review mentions of a query word across an archive. For instance, CLIOQUERY includes a novel NLP-based textual summary feature, that presents a synopsis of all occurrences of a query word in a corpus (Section 4.3).

Focusing on query words in context is a departure from prior work in text analytics, which emphasizes other latent and observable textual units of analysis, such as topics [72], events [76], document metadata [38], interrelated entities [49] or thematic hierarchies [12]. It is also a departure from keyword document search systems [95, 96], which focus on guiding people to ranked documents (Section 2.2.1).

Our atypical decision to design and build a text analytics system for analyzing occurrences of query words across a corpus followed from our systematic investigation into the tasks and expectations of historians and archivists (Section 3), who often review queries in text. Heer et al.'s highly-cited guidance for text analytics [21] stresses the importance of choosing units of analysis which are best "aligned" to the "tasks, expectations and background knowledge" of intended users.

Because CLIOQUERY uses query words in context as its central unit of analysis, the system differs from prior tools in several key ways. We highlight the most important differences below.

2.3.1 CLIOQUERY displays query words in underlying text. Unlike CLIOQUERY, some prior systems for interactive text analysis are designed for analyzing high-level textual units such as corpus themes or temporal trends. As a result, these systems sometimes do not allow people to review occurrences of query words in underlying documents. For instance, visual summaries like the Word Tree [136] and Phrase Net visualizations [132], or time series displays like Theme River [56] do not show query words in underlying text. Similarly, some text analytics systems, such as early versions of Overview [12], experimented with alternatives to query-based paradigms. By contrast, CLIOQUERY is designed to help users find and read query words in underlying documents; the tool's central units of analysis (i.e. query words in context) are spans of text from documents in the corpus (see Section 3.1). This design choice was informed by prior work, which emphasizes the importance of displaying underlying text in interactive text analysis (see Section 9.2).

2.3.2 CLIOQUERY offers complete access to all query mentions in context, without extraneous mediating abstractions. Like CLIOQUERY, some text analysis tools do include features to help people navigate to query words in context. However, because these systems are chiefly designed for

⁷The Overview authors describe the importance of adding query features in discussing their work [12, 124].

analyzing other textual units (e.g. topics [72], events [76] or document hierarchies [12]), they offer only indirect and incomplete access to query words in underlying text, via extraneous mediating abstractions. By contrast, CLIOQUERY is designed to help people directly review all occurrences of a query in a corpus.

 For instance, EventRiver [76] helps people review temporal document clusters, which serve as the system's primary unit of analysis. In principle, a historian could use EventRiver to find and analyze query mentions in context by (1) finding document clusters containing a query word Q using the tool's search-by-keyword feature, (2) clicking such clusters and (3) using the tool's Shoebox and Storyboard features to review those occurrences of Q which happen to fall within documents from selected clusters. However, such a workflow would have two downsides. First, the workflow would be *indirect*; the historian would have to navigate through clusters to access query words in context. Such indirect navigation would force the historian to attend to what Heer *et al.* describe as "extraneous information that might confuse or hamper interpretation" [21]. Second, the workflow would be *incomplete*; a historian would have no way to navigate to occurrences of query words which do not happen to fall within algorithmically-defined clusters. As we describe in Section 3, this would likely pose a problem for historians, who need to directly and comprehensively observe all mentions of their query in context, with minimal confounding algorithmic influence.

Our focus on EventRiver merely serves as one illustrative example of a broader phenomenon. TIARA's mediating topic abstraction [72], Overview's mediating hierarchy abstraction [12], Star-Spire's mediating cluster abstraction [11] and Jigsaw's mediating entity abstraction [49] (some queries are not entities, e.g. "race suicide" [79]) would also force historians and archivists to navigate to query mentions in context via similarly confounding and extraneous abstractions.

2.3.3 CLIOQUERY employs query-focused summarization to ease the burden of reading and context switching. Like CLIOQUERY, some text analytics systems include a snippet feature which shows words extracted from documents in a corpus. Examples include TIARA [72] Snippets, the Overview and Footprints Document List [12, 65], and results snippets from keyword document search systems [28, Chp. 6.3]. While such snippets are visually-similar to CLIOQUERY'S Document Feed (Section 4.3), they differ in important ways, which are crucial to the work of historians and archivists.

Most importantly, many existing snippet components select and display only some occurrences of a query in a corpus. For instance, TIARA's Snippets feature displays a selection of occurrences of a topic word in underlying documents [72], and the Footprints [65] Document List displays the first sentence in a document, regardless of whether the sentence contains a query word. Similarly, keyword document search systems create and display snippets based on heterogeneous criteria [28, Chp. 6], rather than display all occurrences of a query word in ranked documents. Prior systems also do not attempt to help people understand how text is selected for a snippet. For instance, the Overview Document List [12] selects and displays keywords based on an opaque clustering algorithm.

These properties make prior snippet features poorly suited to historians, who need to review all occurrences of a query word in a corpus, with minimal confounding algorithmic influence. Therefore, instead of relying on prior snippet features, a historian who wished to review all occurrences of a query word in a corpus would likely have to click through from snippets to underlying documents, which are often [49, 72, 95] shown in individual windows or tabs. In Section 1 and Figure 2, we explain how this workflow imposes unnecessary reading and context switching costs. By contrast, CLIOQUERY's snippet-like Document Feed employs novel query-focused summarization techniques in order to allow historians and archivists to quickly scroll through and examine every single occurrence of a query term in a corpus. We offer qualitative and quantitative evidence of the importance of this feature in Sections 6, 7 and 8.

ACM Article No:12 Anon.

3 NEEDFINDING STUDY: DEFINING PRACTICES AND REQUIREMENTS

3.1 Formalizing historians' current practice of mention gathering and analysis

In Section 1, we document and informally describe historians' current practice of mention gathering and mention analysis. We now define this work more formally. During mention gathering, a historian investigates a unigram query Q in a newspaper archive. Q is a word type and each mention of the query, $i \in \mathcal{M}(Q)$, is a word token. For instance, a historian might investigate Q="Falluja" by gathering specific mentions of the word "Falluja" in individual documents, published on particular dates. Using d to refer to the text of a specific document and t to refer to its publication date, we can formally define mention gathering as the task of finding all $\mathcal{M}(Q)$ in an archive $\mathcal{H}=\{(d_i,t_i),(d_k,t_k)...(d_N,t_N)\}$, which is an unordered set of N timestamped documents.

The task of mention analysis consists of manually reviewing one or more query mentions in context. We use the notation C(i) to refer to a specific passage showing a particular query mention in context, where C(i) is a token span. For instance, if "Falluja" occurs in document d, then C(i) might be a paragraph from d that contains the string "Falluja". We denote this using $C_{\text{paragraph}}(i)$. Note that different systems may define C(i) in different ways. For example, keyword document search systems return whole documents. Thus a keyword document search system defines C(i) as the whole document d containing $i \in \mathcal{M}(Q)$; we denote this using $C_{\text{full doc.}}(i)$.

Having now formally defined historians' current practice of mention gathering and mention analysis, and explained the limitations of baseline tools for these tasks (Section 1), we now describe an investigation into the needs of historians (Section 3.2) which informs our design requirements for a text analytics system (Section 3.3).

3.2 Observing and analyzing needs from heterogeneous data

We identified historians' needs by collecting and analyzing two different sources of data, described below.

3.2.1 Observing needs from existing literature. First, we studied historians' needs by reviewing a large literature from history, library science and information science devoted to the systematic study of the digital and non-digital information-seeking behavior of historians. To identify this literature, we followed citations starting from Allen and Sieczkiewicz's paper "How Historians use Historical Newspapers" [2], which we first found via a search on Google Scholar. In total, we reviewed and took notes on six prior studies describing surveys and interviews with 1002 historians (shown in a table in the Appendix). We consider our synthesis of this prior literature to be part of the contribution of our work, as we translate these prior descriptive findings (focused on how historians find information) into actionable design requirements for an interface. The studies we review are largely unknown in computer science disciplines like NLP, IR, VIS and HCI.

3.2.2 Observing needs from interviews and feedback on prototypes. We additionally supplemented, contextualized and validated existing studies by conducting five of our own one-on-one needfinding interviews with five interviewees (I1 to I5) on Zoom video chat over a period of three months. The Appendix describes the backgrounds of interviewees in detail. All but one interview was 60 minutes long. (We met with I4 for 30 minutes, due to limited availability.) Interviews proceeded in two phases. During *Phase A*, in the initial exploratory stage of our work, one researcher from our group interviewed I2, I4 and I5, who we recruited through convenience sampling [47]. The interviewer asked open-ended, exploratory questions about needs and practices, and solicited feedback on

⁸ All three of our human studies—needfinding interviews, expert interviews, and field study (Sections 3.2, 5, and 7 respectively)—were approved as exempt from review by our institution's human subjects IRB office. All participants received a \$50 Amazon gift card for their time.

 early prototypes. The researcher also took detailed notes. Later, when we better understood how historians find information in archives, we began *Phase B*. During this phase, the same researcher conducted two one-on-one, video-recorded, semi-structured interviews with I3 and I1, who also provided feedback on later prototypes. We recruited I3 and I1 via email outreach. The researcher again took detailed notes. We include the interview script in supplemental material. In total, each of the five interviewees across *Phase A* and *Phase B* reviewed a different iterative prototype. In the interest of space, we only present feedback on what we consider to be the two most important prototypes, shown in the Appendix.

3.2.3 Analyzing observations of historians' needs. Following data collection, one researcher qualitatively analyzed and organized notes and transcripts to articulate four overall needs, and translate these needs into four corresponding design requirements (described in Section 3.3). In general, we found that feedback from needfinding interviews and feedback on early prototypes was very consistent with findings from prior work. Nevertheless, our own needfinding interviews helped to contextualize and translate prior descriptive findings on historians' information-seeking behaviors into actionable guidelines for system design.

3.3 Needfinding results and design requirements

Following data collection and data analysis, we defined four high-level design requirements (R1-R4), based on four needs. We describe each requirement below.

3.3.1 R1: A system should show a navigable overview of change over time. Prior study of the information-seeking behavior of historians emphasizes the theoretical importance of "the dimension of time" [16] in historical research, and also emphasizes historians' practical need to perform "searching and narrowing by date" [2]. In our needfinding interviews, historians and archivists also stressed the theoretical and practical importance of time-based investigation. "Time is always a historian's first move," I3 explained. "It's about change over time as the fundamental thing." I5 noted: "Historians are often trying to find articles within a specific date range and about a specific topic ... research often starts with a keyword and a date range and a source or list of sources." Because historical research involves studying change across time, I2 explained how time series plots showing the frequency of query words by time period (see Figure 3c) are often useful for gaining a temporal overview of a corpus. "Bar charts [or line charts] by time are really helpful," I2 explained, "because news has these peaks where a topic becomes important and then dies down." Such charts "help people trace an idea or series of ideas or terminology over time." Observing the centrality of temporal analysis in historical research, we assert a design requirement (R1): a system designed for historical mention gathering and analysis should show some kind of navigable, visual overview of query mentions $\mathcal{M}(Q)$ across the time span of a corpus. Showing such a visual "overview first" [58, 118] is a known best practice in visual analytics.

3.3.2 R2: A system should help people comprehensively review all query mentions in a corpus. Prior work often emphasizes the importance of gathering comprehensive evidence during historical research. "Comprehensiveness is clearly the highest priority in searching a database," one study concludes [31], explaining that 70% of 278 survey respondents would prefer to spend time filtering out irrelevant material than run the risk that relevant material "might fall through the cracks" in a limited search. Nevertheless, some historians in prior work acknowledge that truly

⁹We emailed five PhD students in history at a nearby university. Each student expressed interest in media, archives or science in describing their work on their department's web page. We also emailed all members of the editorial board at a history journal. We do not list the name of the university or journal to ensure interviewees remain anonymous.

ACM Article No:14 Anon.

comprehensive search is an impossible goal. "I never think I'm going to be able to read every record..." one reports [36]. "I'm always creating priority orders of what I think is going to be most useful."

Our interviewees similarly emphasized the importance of comprehensiveness in gathering and evaluating historical evidence. "The most important thing for historical researchers is to be confident that they are being exhaustive" said I4. "I want to know I can be confident I have been able to access everything relevant. Did my search cast a wide enough net?" I4 also praised an early prototype (see Appendix) for displaying a very large number of potentially relevant passages. "The biggest fear is Type II error," he explained. "In doing searches, am I missing something that is crucial but I don't know because I never looked"? Similarly, I5 explained that it is important to "be as completist as possible" in historical research. "The thing about historians....they want to be as comprehensive as possible with their topics." Citing the importance of comprehensiveness, I5 expressed deep skepticism (see Appendix) about an early prototype which omitted some information to form a summary. However, like some interviewees in prior work, I2 pointed out that truly comprehensive investigation may not be possible. "Ultimately," she noted, "there is a limit in terms of time and money for any given project". We translate the need for comprehensive archival search into a second design requirement (R2): a system for mention gathering and analysis should help people comprehensively review all query mentions in a corpus. Expressed more formally, a system should help historians easily navigate to and review every single $i \in \mathcal{M}(Q)$ in an archive.

3.3.3 R3: A system should present as much context as possible for any given record in an archive. Prior work emphasizes the importance of context in historical research. "Building context is the sine qua non [indispensable condition] of historical research," Duff and Johnson write [36]. "Without it historians are unable to understand or interpret the events or activities they are examining." Similarly, another historian explains, in a separate study, "You can't have the specific facts without the context ... Where an article is in the paper, and what surrounds it, matters."

During our own needfinding interviews, historians and archivists also repeatedly emphasized the importance of contextual information in archive news search. The job of a historian is to "put facts in context" I5 said. A historian will need to "contextualize" facts from a periodical by examining its publishers and audience. Similarly, I4 noted that "as an archivist I do research to give context to collections." Finally, I2 stressed the importance of contextualizing evidence in archive search software. "Who does the New York Times have writing this?" I2 asked, while examining an early CLIOQUERY prototype. "Where does each sentence occur in the document? What section of the newspaper? You need to show more context".

Observing the importance of context in historical research, we assert a design requirement (R3): a system for mention gathering and analysis should show each query mention amid as much surrounding context as possible. Formally, R3 implies that |C(i)| should be as large as possible for each $i \in \mathcal{M}(Q)$. However, note that R3 must be balanced against other requirements, which impose competing demands. In particular, because screen space and human attention are limited resources, showing a lot of context around an individual mention (e.g. showing the mention within the context of a full document) will make it harder for a historian to comprehensively review all mentions in a corpus. ClioQuery considers how to balance the need for context and comprehensiveness.

3.3.4 **R4:** A system should be as transparent, trustworthy and neutral as possible. Prior studies of the information-seeking behavior of historians underscore the need for trustworthy tools that transparently present digital archival materials in a neutral manner. For instance, in one study [35], historians report that each digital source must be "accurate, undistorted and complete," explaining that each source is "not edited in some way." Similarly, in another study [17], a historian explains that direct "access to the original image of the primary source rather than to a transcribed

version" is important "especially when there is no description of what rules they used to transcribe documents". This historian reports that they do not trust and can not interpret electronic transcription, and thus must rely on direct observation of digitized images to draw conclusions.

In our interviews, historians and archivists similarly described the importance of transparently presenting digitized archives in a neutral manner. "When I see something that is trying to decide or curate for me that is a worry. That is a red flag," I4 explained. Similarly, I2 added, "I think the system should be as transparent as possible. I need to distinguish between what some primary source is saying versus what the computer thinks a primary source is saying." I5 also cited the importance of transparency and trust in expressing deep skepticism about an early prototype, shown in the Appendix. Because historians frequently expressed commitments to direct and neutral observation of archive evidence, in designing new archive tools, we assert a design requirement (R4): search software should show archive evidence in a maximally transparent and trustworthy manner. One consequence of R4 is that systems should not attempt to create a curated summary of the most "important" $\mathcal{M}(Q)$ in an archive (see Section 9.3).

4 SYSTEM

 CLIOQUERY is a visual analytics system designed to support historians in their practice of mention gathering and analysis. The system is unique within a large literature on text analytics because it is designed to help analyze query mentions in context, which constitute the system's central "unit of analysis" (this terminology comes from Heer *et al.* [21]). By contrast, prior text analytics tools focus on the investigation of other units of analysis like topics [72], events [76] or thematic hierarchies [12] (see Section 2.3). Similarly, keyword document search tools are organized around ranking whole documents; they do not focus on the analysis of query mentions within documents.

CLIOQUERY's unique focus on query mentions in context reflects our systematic study into the needs and practices of historical researchers, who helped refine early iterative prototypes [50]. Because such historians need to quickly and comprehensively review occurrences of a query word in a large corpus, CLIOQUERY uses text simplification techniques (Section 4.7) to create a skimmable summary of a query across an archive. Such text simplification techniques and skimmable summaries are unique within the large literature on text analytics, and thus constitute the chief technical contribution of our work. We hypothesize that CLIOQUERY query-focused summarization, along with its linked views, in-text highlighting, and history tracking features, can together help experts quickly, comprehensively and transparently gather and analyze the comprehensive set of all mentions $\mathcal{M}(Q)$ of a query in a news archive.

4.1 High-level system description

The CLIOQUERY web interface presents results from a Boolean search [80, Chapter 1], which returns the unranked set of documents containing one or more mentions $\mathcal{M}(Q)$ of a unigram query term Q in an archive \mathcal{A} . (This notation is defined in Section 3.1; Section 10 discusses possible extensions to exact string matching.) When a user enters Q into the search bar at the top of the interface (Figure 1A), CLIOQUERY identifies all documents containing Q and presents the documents using three linked views [15]. First, CLIOQUERY includes a **Time Series View**, showing a graphical overview of the count of documents mentioning the query by year (Figure 1D). Second, CLIOQUERY includes a **Document Feed** view, presenting all query mentions from across all documents in a single scrollable window (Figure 1H). Finally, CLIOQUERY includes a **Document Viewer**, which shows

¹⁰Even as some interviewees stressed the importance of unbiased, transparent and trustworthy presentation of archive evidence, I3 reported that, in practice, historical researchers do trust ranked results from keyword document search systems. She explained that many historians might not realize that black-box document rankings from a keyword document search tool will affect conclusions from archival research.

ACM Article No:16 Anon.

the full text of a single document from the corpus, with individual query mentions from the document highlighted in context (Figure 1I). CLIOQUERY also includes a **filtering system** to help users narrow the set of query mentions shown in the interface (Figure 1B, C and F), and a **history tracking system** to automatically monitor and display reading history during comprehensive search (Figure 1G). All features in the interface also follow a coordinated **color coding** scheme. For instance, the user's query word is always displayed using the purple query color in the Document Feed and Document Viewer, and the Time Series View also uses a purple line to represent query frequency (Figure 1D). We consider the color-coded bolding of query terms to be one form of **automatic in-text highlighting** [52] throughout the CLIOQUERY interface. Automatic in-text highlighting draws user attention to some word, phrase or passage in text by automatically setting the text's foreground color, background color or decoration (e.g. bolding). The Appendix describes our process for selecting a colorblind safe and print-friendly palette. It also provides additional engineering details about our implementation of CLIOQUERY.

4.2 Overview first: a Time Series View for temporal context (R1)

 Because change across time is central to historical research (R1), CLIOQUERY presents a navigable Time Series View (Figure 1D) showing query frequency by year across a corpus. The component's x-axis represents time (binned by year) and the y-axis represents the annual count of all documents containing the query Q published during a given year. If a user also enters a subquery (Section 4.6), CLIOQUERY'S Time Series View also shows the annual count of documents mentioning both the query and subquery. In Figure 1D, CLIOQUERY displays one line showing the count of documents mentioning the query term in the purple query color, and another line showing the count of documents mentioning the subquery term (as well as the query term) in the green subquery color CLIOQUERY'S time series plot also shows a single rug point (small vertical line) for each document mentioning the query, just beneath the temporal x-axis (Figure 1E). Such rug points allow the user to easily preview and navigate to individual news stories; we describe these possible interactions in detail in the Appendix.

4.3 A Document Feed for comprehensive search (R2)

During needfinding, we found that experts often emphasized the importance of gathering comprehensive evidence (Section 3.3.2), and also often search for specific query terms in news archives (Section 1). We thus designed CLIOQUERY'S Document Feed to help such users easily gather and analyze the comprehensive set of every single mention of a query term in a collection of news stories (R2). We assume the user is working with a small corpus (or small set of documents from a larger corpus), where such comprehensive review is possible. This assumption is appropriate for our use case; for instance, Black reviews roughly 500 documents to analyze the racial history of "watermelon" [8] and MacNamara reviews 605 documents to analyze "race suicide" [79].

After a user issues a query Q, CLIOQUERY populates the Document Feed to show a comprehensive, skimmable, summary which includes every single $i \in \mathcal{M}(Q)$ across the corpus (Figure 1J). To create the summary, CLIOQUERY selects each sentence containing some $i \in \mathcal{M}(Q)$, and then automatically simplifies the sentence (without removing query words) so that historians can quickly read over the mention i in context. We use the notation $C_{s'}(i)$ to refer to a specific query mention $i \in \mathcal{M}(Q)$ shown within the context of a sentence s that is simplified to s' for display in the Document Feed. Section 4.7.1 provides details on how CLIOQUERY shortens sentences to create $C_{s'}(i)$. To the best our knowledge, such text simplification is new to the literature on text analytics. In presenting each $C_{s'}(i)$, CLIOQUERY bolds and highlights the query word using the purple query color, so that each $C_{s'}(i)$ is shown in a visually consistent format designed for skimming (Figure 4).

Context	Num. tokens
$\overline{C_{\mathrm{full doc.}}}$	222,544
$C_{\mathrm{full\ sent.}}$	49,382
C_s	28,859

Table 3. Total tokens presented to the user, if each mention of the query "Reagan" and subquery "Duarte" is shown within the context of a full document ($C_{\text{full doc.}}$), a full sentence ($C_{\text{full sent.}}$) or a shortened sentence (C_{s}). By shortening sentences, CLIOQUERY removes 87.0% of tokens from all documents containing a query mention, and 41.6% of tokens from all sentences containing a query mention. Counts come from the example in Figure 1, using the Salvador corpus (see Appendix); we assume the Document Feed is sized to fit within half the width of a 13-inch screen. [Abe: minor: Table 5 might very slightly benefit from having another column that's the reduction-in-reading-time ratio compared to the $C_{fulldoc}$ baseline. To do this need to make numbers reported in caption consistent as fraction of whole doc.]

Note that by default, CLIOQUERY displays a single $C_{s'}(i)$ from each document beneath the document's headline (The Appendix describes how the sentence is chosen). To see each $i \in \mathcal{M}(Q)$ from a document within a simplified sentence, the user can click an "expand" button (Figure 2). The user can also click a star to bookmark a document in the red bookmark color ...



Fig. 4. CLIOQUERY's Document Feed uses text simplification methods from natural language processing in conjunction with color-coded, automatic, in-text highlighting, in order to summarize query mentions $\mathcal{M}(Q)$ in context, in a visually consistent format designed for skimming. Here one portion of one document (top of stack, left) containing three mentions of the query term Q="Georges" has been shortened into a summary of "Georges" (right). To create this summary, CLIOQUERY extracts each of the three sentences mentioning "Georges" and simplifies each sentence to render a shorter sentence $C_{S'}(i)$ that is fewer than 60 characters long. In this figure, for illustration, spans of tokens from one source document included in the summary are shown with italicized and underlined text. In typical use of CLIOQUERY there are often hundreds or thousands of documents containing Q (shown as document stack, above left).

CLIOQUERY's Document Feed is designed to directly address two of the limitations of baseline keyword document search systems, described in Section 1. First, by summarizing documents mentioning Q, CLIOQUERY is able to fit more query mentions in limited screen space, reducing the need for context switching across individual windows or tabs. For instance, in Figure 1, the Document Feed saves the user from having to open 239 separate documents during comprehensive review. Second, by selecting sentences mentioning the query from documents, and removing tokens from those sentences, CLIOQUERY reduces the user's reading burden. For instance, in Figure 1, the user has queried for documents mentioning Reagan and Duarte (in this example, Reagan is a subquery; subqueries are described in detail in Section 4.6). By selecting and simplifying sentences, CLIOQUERY removes 87.0% of the tokens in all documents mentioning these two words (Table 3). We include a detailed description of CLIOQUERY's text simplification techniques in Section 4.7.

ACM Article No:18 Anon.

4.4 A linked Document Viewer for necessary context (R3, R4)

 Because historians need to evaluate evidence in context without black-box algorithmic influence (R3, R4), we anticipated that CLIOQUERY users would need to quickly review each $i \in \mathcal{M}(Q)$ from the Document Feed within the context of full underlying news articles. Therefore CLIOQUERY'S Document Feed is closely linked with a corresponding **Document Viewer**, which shows the complete text of a single selected document from the corpus (Figure 1I). The Document Viewer satisfies R3 because it shows each $i \in \mathcal{M}(Q)$ within the context of a full document, denoted $C_{\text{full doc.}}(i)$. After a user clicks a shortened sentence $C_{s'}(i)$ in the Document Feed, the Document Viewer updates to show the entire document containing $C_{s'}(i)$. CLIOQUERY also automatically scrolls the document so that the (just clicked) simplified sentence is visible on screen.

CLIOQUERY also makes it easy for users to locate simplified sentences, by using automatic in-text highlighting to further link the Document Feed and Document Viewer. Each simplified sentence $C_{s'}(i)$ from the Document Feed is shown with yellow background highlighting \square in the document shown in the Document Viewer. Additionally, if a user hovers over a sentence in the Document Feed or Document Viewer, the sentence is highlighted in dark yellow hover color \square in each component (shown in Figure 1J and 1K). We hypothesize that linking between shortened text and full documents helps build user trust (R4), because it helps experts transparently see and understand how shortened mentions are drawn from underlying text. This feature is inspired by CommunityClick [66].

4.5 Color-coded history tracking for systematic review of evidence (R2)

Some historical researchers emphasize the importance of comprehensively examining all available evidence during research (R2). To support historians in this work, CLIOQUERY keeps track of which documents the analyst clicks in the Document Feed and opens in the Document Viewer. CLIOQUERY also keeps track of bookmarked news stories (Figure 1J), and displays a simple stacked horizontal bar chart (Figure 1G) showing the proportions and counts of read, unread and bookmarked documents. The bar chart uses the read \square , unread \square , and bookmarked \square color scheme employed across the color-coordinated interface. (CLIOQUERY considers all documents to be either read but not bookmarked, unread or bookmarked. We do not allow intersection between these sets.) For instance, Figure 1G shows 5 read, 89 unread and 5 bookmarked documents. The user can click check marks (Figure 1G) to show or hide documents in each category.

CLIOQUERY'S Document Feed and Time Series View use the same color scheme to help users quickly identify opened and unopened documents. Stories that a user has already clicked appear with grey read text in the Document Feed, and their corresponding rug points are shown in grey in the Time Series View. For instance, in Figure 1, the user has read the story published on Jan. 9, 1985. The story is greyed out in the Document Feed, and its corresponding rug point is shown in grey beneath the time series plot. Similarly, there are five red rug points in Figure 1E, because the user has bookmarked five documents.

Note that CLIOQUERY's history tracking is query-dependent; tracking resets each time a user issues a new query, unlike the history tracking mechanism in some prior work [65, Section 6]. Such query-dependent tracking is appropriate for CLIOQUERY, because the system is designed to help historians review all mentions of some specific keyword in a corpus. We hypothesize that this feature offers experts assurance they have comprehensively reviewed all documents containing some $i \in \mathcal{M}(Q)$, and leave exploration of other forms of history tracking for future work.

4.6 A filtering system to review many results in a neutral manner (R4)

Some prior text analysis systems designed for historians (e.g. Expedition [119]) attempt to answer keyword queries by ranking documents to direct users towards most-relevant news articles. Because

such ranked retrieval might introduce unwanted algorithmic influence over the expert search process (R4), CLIOQUERY instead responds to queries with Boolean search, which returns the unranked set of all documents mentioning a query term. (The Document Feed shows such documents in chronological order.) CLIOQUERY then allows users to narrow down unranked search results with a filtering system, consisting of three filter controls.

 The **filter-by-date** control selects documents by time period. After users select a start date and end date from date pickers at the top of the interface (Figure 1B), CLIOQUERY updates to show only those documents mentioning the query published during the selected interval. (Historians are often interested in specific time periods; see Section 3.) In Figure 1B, the user has filtered to documents published in 1983–1985.

The **filter-by-subquery** control allows users to select documents that mention some additional word, called a subquery. For instance, after a user queries for the Salvadoran leader "Duarte" they might wish to further narrow results to understand the relationship between "Duarte" and his ally U.S. President Ronald Reagan. To investigate this question, the user can enter the subquery "Reagan" to select all documents mentioning the word "Duarte" which also mention the query word "Reagan" (Figure 1C). We included this feature because complex Boolean queries are often popular with experts [80, Section 1.4]. More complex Boolean expressions are possible in future work.

The **filter-by-count** control filters results based on the the number of times a query term is mentioned in a document. When a user adjusts the filter-by-count slider to some value $K \in \{1, 2, 3, 4, 5\}$, all components of the interface update to show only those documents with K or more $i \in \mathcal{M}(Q)$. In cases where a user has set a subquery, the filter-by-count control allows the user to select documents which contain the subquery word at least K times. For instance, in Figure 1F, the user selects documents which mention "Reagan" at least 3 times.

CLIOQUERY also helps users quickly see the count of query terms within documents using square-shaped, query-colored **count markers**, shown beside each document headline. Count markers use brightness to encode the count of a query term within a document. For instance, count markers for documents with more mentions of a query term have a darker purple color than count markers for documents with fewer mentions. If a user enters a subquery, count markers show the count of the subquery within each document, using shades of the subquery color (as in Figure 1F and 1H). This feature is inspired by TileBars [57].

4.7 Sentence simplification to help summarize a query across a corpus

CLIOQUERY introduces text simplification methods from NLP to the literature on text analytics. We describe these methods below.

4.7.1 Overview of sentence simplification in CLIOQUERY. CLIOQUERY'S Document Feed displays a query-focused summary of a user's query and subquery, by first extracting and then simplifying sentences mentioning query (or subquery) words. To simplify sentences, we turn to sentence compression techniques from the text summarization literature in NLP (introduced in Section 2.1.2). These methods try to summarize naturally occurring input sentences by removing words, to create shorter and well-formed output sentences, which contain the most salient information from the input. (A well-formed sentence is one that sounds natural, rather than garbled or choppy [113].) In particular, we turn to a specific class of sentence compression methods, which can ensure that simplified sentences both (A) fit within limited screen space in a user interface and (B) mention the user's query term or subquery term. Such methods are appropriate for CLIOQUERY because each line in the Document Feed has a fixed width, and must include some mention of the user's query or subquery.

ACM Article No:20 Anon.

More concretely, we use a *query-focused clause deletion* [53, 55] method to shorten sentences in cases when a user has entered a query (Section 4.7.2), and also use *relationship span extraction* [54] in cases when a user has entered both a query and subquery (Section 4.7.3). We also employ a final fallback approach, *character windowing*, when it is not possible to shorten a sentence using other techniques (Section 4.7.2). In the next sections, we describe each sentence shortening method in greater detail. The Appendix provides additional details on how CLIOQUERY chooses between possible sentence shortening methods. ¹¹

4.7.2 Query-focused clause deletion, and character windowing. CLIOQUERY's Document Feed requires shortened sentences that mention Q and fit within available screen space. We assume that such shortenings should also be well-formed and contain the most salient information from longer source sentences. Prior research in IR suggests that users prefer well-formed snippets [22], and prior work in sentence compression [42, 43, 70] strives for both well-formedness and salience. We also assume that methods for constructing shortenings must run with low latency, which is known to be important in user-facing analytics systems [73]. Different sentence shortening techniques might optimize for and manage tradeoffs between such requirements. But in this work we turn to a simple query-focused clause deletion method to meet such criteria, allowing us to focus on how to apply text summarization methods in user interfaces for historical research.

Query-focused clause deletion exploits the fact that natural language sentences are sequences of words, which exhibit hierarchical and nested grammatical structure [7]. For instance, the sequence "She swims in the pool" can be divided into interrelated word groups, with specific grammatical relationships; the words "in the pool" form a prepositional phrase that modifies the verb "swims." To represent such linguistic structure, clause deletion employs a dependency parse tree [98] grammatical formalism. A dependency parse is a directed tree graph with one vertex for each word in the sentence, along with a latent root vertex. ¹² Each subtree in the parse corresponds to a constituent subsequence in the sentence. The sentence simplification literature sometimes describes such subtrees as *clauses* [44]. Figure 5a shows an example dependency parse.

Sentence simplification via clause deletion shortens sentences by iteratively deleting clauses from a dependency parse. Figure 5 shows how one sentence is shortened by iteratively deleting two clauses. Unlike sentence compression techniques which consider individual tokens for removal (e.g. Filippova $et\ al.\ [42]$), deleting clauses naturally identifies and removes groups of related words. For example, a single deletion could remove the prepositional phrase "after the election," or a much longer word group with more modifiers and embedded clauses: "after the previous election last year, which went poorly." Shortening sentences via clause deletion also makes it easy to ensure that output sentences must include Q; clauses that contain query mentions are not allowed to be removed during deletion. 14

To try and create well-formed output sentences, CLIOQUERY turns to prior work on clause deletion [53, Section 6], which has found that in general removing more clauses from an input sentence makes it less likely that the resulting output sentence will be well-formed. Thus, to shorten an input sentence, CLIOQUERY's clause deletion first identifies those candidate output shortenings

¹¹In Figure 1, CLIOQUERY uses relationship span extraction to shorten and display some sentence from 31 out of 239 documents which mention "Duarte" and "Reagan". It uses query-focused clause deletion to shorten and display some sentence from 85 documents, and it resorts to character windowing for the remaining 123 documents.

¹²We use the UD (v1) dependency formalism [98]; other related formalisms allow for non-tree parses [112]. Eisenstein [40, Chapter 11] offers a broad introduction to dependencies. We perform dependency parsing using Stanford CoreNLP [18, 81].

¹³Tokens from the remaining tree are then printed in left-to-right order, based on their position in the original sentence.

¹⁴It is also possible to enforce such query constraints using integer linear programming (ILP). However, ILP-based sentence

¹⁴It is also possible to enforce such query constraints using integer linear programming (ILP). However, ILP-based sentence compression techniques (e.g. Clarke and Lapata [23]) are NP-hard and have been shown to be orders of magnitude slower than other iterative approaches to query-focused sentence compression [55].

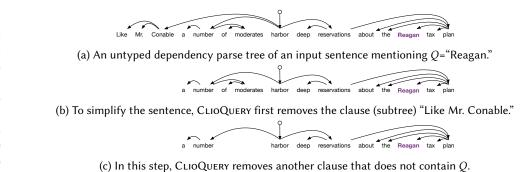


Fig. 5. Sentence simplification via query-focused clause deletion [53, 55]. CLIOQUERY removes two subtrees from a dependency parse across two steps to simplify the input sentence (5a) into the output sentence (5c).

that can be constructed by removing at most K clauses from the input (without removing Q), and are also short enough to fit in one line of text within the Document Feed. Because in practice it is often possible to dramatically shorten an English news sentence by removing only one or two large clauses (for example, a lengthy relative clause, such as "Reagan met with the envoy who was sent by the ..."), CLIOQUERY only considers shortenings which can be constructed by removing $0 < K \le 2$ deletions. ¹⁵

To try and ensure that output shortenings include the most salient information from input sentences, CLIOQUERY then returns the candidate output shortening with the highest tf-idf score [80]. Tf-idf scores are often used in extractive sentence compression [23, 44] and text summarization [32] to identify salient information for inclusion in summary output; the metric identifies words which occur with unusual frequency (relative to the overall corpus), which is an important signal of salience in summarization [94]. The Appendix includes details of how we compute td-idf in CLIOQUERY to identify words which occur frequently in documents mentioning a query.

In some cases, there is no way to shorten a sentence by removing one or two clauses while ensuring that that output sentence mentions Q and will fit in the Document Feed. In these circumstances, ClioQuery resorts to shortening the sentence by extracting the span of N characters to the left and right of Q in the sentence, where we maximize N under the constraint that the resulting character span will both fit in Document Feed and respect word boundaries. We use this *character windowing* method only as a last resort because it may cut off syntactic constituents (e.g. show only a portion of a prepositional phrase), which may create awkward-sounding output. Footnote 11 describes how often CLIOQUERY uses this fallback, during an example run of CLIOQUERY.

In the future, it might be possible to shorten more sentences with query-focused clause deletion by considering candidate output shortenings that are created using more than K=2 deletions. (Prior work on query-focused clause deletion does not yet offer an efficient solution for considering such candidates [53].) Because the number of candidates grows with K, developing algorithms

¹⁵In addition to encouraging well-formed output, this strict limit ensures low latency for the user. For a sentence M words long, the worst case for performance is a tree where all words are leaf vertexes, resulting in M + M(M-1)/2 possible outputs of K = 1 or 2 deletions. But in typical trees, there are far fewer possible deletions because: (1) the query word and all its ancestors are not allowed to be deleted, (2) after the first deletion of a clause length C (i.e., the size of the deleted subtree) only M - C candidates remain for the second deletion, and (3) if CLIOQUERY finds any candidate shortenings using K = 1, it won't search for candidates using K = 2, as shortenings which remove fewer clauses are more likely to be well-formed. We do not consider cases where K = 0, as most unshortened news sentences are too long to fit within the Document Viewer.

ACM Article No:22 Anon.

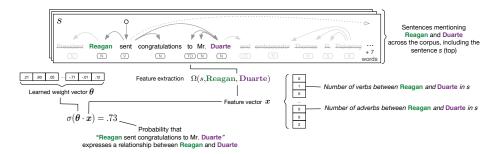


Fig. 6. Sentence simplification via relationship span extraction [54]. This method predicts the probability that the span of tokens between the query and subquery words in a given sentence s will sound natural as a shorter and standalone sentence. (If the predicted probability is high, the sentence can likely be shortened to the span of tokens.) This figure shows the predicted probability that the token span between the query **Duarte** and the subquery **Reagan** will sound natural as a shortened sentence, when extracted from the longer sentence shown in Figure 1 (letter K). Seven words from s are not shown in the diagram above.

which efficiently search over possible outputs or learn greedy deletion policies based on data (e.g. with reinforcement learning) might offer useful starting points.

4.7.3 Relationship span extraction. CLIOQUERY users who search for a query term Q can also filter query results by a subquery. When a user enters both a query and a subquery term, we assume that they are broadly interested in how these two terms are related in the corpus. For instance, a user might query for the Salvadoran leader Q="Duarte" and apply a subquery for the then U.S. President "Reagan," in order to understand Duarte's relationship to Reagan (Figure 1).

To meet this information need, CLIOQUERY attempts to simplify long and complex sentences mentioning both the query and subquery terms into short sentences which concisely describe the relationship between the query and subquery. We describe the process of shortening sentences in this manner as *relationship span extraction*, because each shortened sentence is a token span (i.e. sequence of tokens) extracted from a longer sentence. For instance, in Figure 1, we extract the span "Reagan sent congratulations to Mr. Duarte" from the longer sentence "President Reagan sent congratulations to Mr. Duarte and Ambassador Thomas R. Pickering pledged United States support for further meetings."

CLIOQUERY relies on a known natural language processing technique to perform relationship span extraction, which is specified in detail in prior literature [54, Sec. 4]. At a high-level, this method employs logistic regression to determine if an input sentence s containing two input query words can be shortened to express a relationship between those two query words. To make this determination, the method first extracts a vector of linguistic features x containing information about the query words in the sentence (e.g. is there a verb token between the query words in s?), and then passes the dot product of x and a learned weight vector θ through a logistic function σ . This returns a predicted probability that the token span between the query and subquery will sound natural when removed from the sentence (Figure 6). In our case, the input query words are the user's query and subquery; we shorten a sentence s to a relationship span if the predicted probability that the span sounds natural is greater than a threshold T=0.5.

In our implementation of CLIOQUERY, as in prior literature [54], relationship span extraction is supervised using a benchmark corpus from Filippova and Altun [43], consisting of single sentences

 $^{^{16}}$ We implement with Scikit-learn [101]. Note that setting a lower threshold T might increase the total number of shortened sentences, at the cost of creating fewer well-formed extractions (and vice versa).

paired with single-sentence summaries, which are automatically generated from news headlines. In principle, a technically-oriented CLIOQUERY user would be able to retrain relationship span extraction on their own corpus, using the technique from Filippova and Altun to automatically generate training data from headlines in their own news archive.

5 EXPERT INTERVIEW STUDY PROCEDURE

After analyzing user needs (Section 3) and designing the CLIOQUERY system based on such needs (Section 4), we conducted an interview study over Zoom video chat to test CLIOQUERY with historians and archivists, who used the system to investigate questions from news archives.

5.1 Recruitment, participants and corpora

We recruited five participants (P1-P5) from two universities in the U.S., by emailing students, faculty and staff listed on history and library department web pages. All participants had advanced degrees (master's or PhD) in history or library science, much like the expected users of our system; we provide more details on the backgrounds of participants in the Appendix. Interviewees from our needfinding study (Section 3) did not participate in our expert interview study, to avoid what Sedlmair *et al.* describe as a potential form of bias [115]. Each participant in the interview study had an established research or curatorial interest in some topic related to late 20th century or early 21st century history, which we express as a single topic word (see Appendix). We identified this designated topic word based on each participant's publication record and professional web presence. Before each interview, we then loaded CLIOQUERY with a corpus of *New York Times* (NYT) editorials¹⁷ published between 1987-2007 [110] mentioning the designated topic word.

5.2 Data collection

 To administer the study, one researcher from our group conducted five, one-on-one, sixty-minute interviews over Zoom video chat. (See supplemental materials for a detailed script.) During each interview, the researcher asked each participant to brainstorm and then articulate a high-level research question, based on the participant's prior work (10 minutes). They then introduced the participant to CLIOQUERY via a tutorial (7 minutes), asked them to investigate their research question using CLIOQUERY (30 minutes) and concluded with a semi-structured interview (13 minutes). Throughout, the researcher observed and recorded participant reactions and invited participants to think-aloud [97] as they used the system. If a participant offered feedback on some portion of the interface during their investigation (e.g. offered detailed feedback on the Time Series View), the researcher did not ask about this topic again during the semi-structured interview.

5.3 Thematic coding

The researcher who conducted the interviews analyzed automatic Zoom transcriptions for each of the video recordings, and corrected transcription errors. The researcher then extracted 183 quotes from across five interview transcripts. Each quote consisted of a few sentences on a focused topic, along with the preceding question or comment to provide context (e.g. a quote might discuss the Document Feed). The researcher attempted to extract as many quotes as possible, excluding irrelevant quotes (e.g. tutorial instructions).

The researcher then developed a codebook of six high-level codes (described in Section 6), by grouping and re-grouping the 183 quotes to identify common themes, much like the codebook-based approach described in Miles, Huberman and Saldaña [91, Chp. 4]. After each assigning each quote

¹⁷Social researchers sometimes study editorials to better understand media sources [19, 75].

 $^{^{18} \}mbox{Miles},$ Huberman and Saldaña [91, Chp. 4] describe assigning codes in two phases; we assign codes in a single phase.

ACM Article No:24 Anon.

to exactly one of the six codes, the researcher shared the codebook with an undergraduate coder with training and experience in qualitative coding, who was not involved with the development of CLIOQUERY. The second coder independently assigned codes to the same quotes, using the codebook. The second coder was also invited to add new codes to the codebook if needed, but reported that no new codes were necessary. (Thus we did not modify the codebook.) We include a copy of the codebook in supplemental materials.

Following independent coding of each of the 183 quotes, the two coders met for 1 hour over Zoom video chat to discuss 41 disagreements, and attempted to reach consensus via discussion. In 21 cases, the two coders were able to reach agreement regarding the appropriate code. In 20 cases, the coders determined that disagreement reflected genuine ambiguity in qualitative data, and agreed to disagree.

McDonald *et al.* [85, Section 2.2] use the term *reliability* to describe the extent to which coders reach the same result from independent work, and use the term *agreement* to describe the extent to which coders reach consensus after discussion. Adopting this terminology, we measure the reliability of the two coders by computing a Cohen's $\kappa = 0.724$ (using the R psych 2.0 package [107]), and we measure the agreement of the two coders by computing a $\kappa = 0.855$.

6 EXPERT INTERVIEW STUDY RESULTS

 In this section, we describe the six themes that emerged from qualitative coding.

6.1 CLIOQUERY helps with historical sensemaking

While using CLIOQUERY, each of the five participants formed a question related to their research and then collected and interpreted evidence to start to answer their question, in a process Dalton and Charnigo [31] describe as historical sensemaking. We observed that CLIOQUERY helped historians in this process, offering partial validation for our hypothesis that CLIOQUERY features can aid historians in their work (Section 4).

For instance, as part of his research, P1 studies *New York Times* news coverage from journalists embedded with United States military units in the Iraqi city of Falluja during the second U.S.–Iraq war. From prior study, P1 understood that embedded U.S. journalists often published news stories reflecting the perspectives of U.S. military leaders. But while examining mentions of "Falluja" in *New York Times* editorials using the CLIOQUERY interface, P1 expressed surprise when finding a more nuanced perspective from the opinion desk. "I didn't see nearly as much of the sort of sensational depiction of Falluja, and the militants in Falluja [in editorials] that I expect from embedded journalists [in news stories] …" he reported.

Similarly, P2 used CLIOQUERY to find confirming evidence of shifting U.S. perspectives towards Robert Mugabe. As P2 expected, early *New York Times* editorials from the corpus praised *Q*="Mugabe" as a liberator, but then began to criticize "him as a bad statesman, as a tyrant and a dictator." P3 was likewise able to partially answer a research question with CLIOQUERY. She explained that while she had "a deep knowledge of [women in combat]. I don't have a deep knowledge of what the [NYT] editorial board has to say about it." Using CLIOQUERY, she found evidence of editorials using "the gendered trope that women are supposed to be wives and mothers." P5 also discovered an unexpected connection with musical copyright, while researching a hypothesis surrounding literary copyright. "The parity [with the music service] Napster that's that's really interesting ... That's not something I thought about ... I was thinking ... definitely more in literary items because that's what I deal with."

6.2 CLIOQUERY features offer a corpus overview, alongside complementary context

Participants offered detailed feedback on CLIOQUERY features during interviews, which often matched our design goals for particular components of the interface. To begin, three participants

 reported that CLIOQUERY'S **Time Series View** offered a useful overview of the entire corpus, by directing their attention to salient time periods. P1 said the Time Series View was an "easy way of visualizing" corpus trends, and P5 suggested that the Time Series View might be helpful "when students are kind of in that exploratory phase ... as a way of ... coming up with research questions." P4 offered similar feedback. "I really like this," she said. "This looks really functional and really useful. I like how there is quite a lot of information packed in."

P1, P2 and P5 reported that the **Document Feed** was a useful feature of CLIOQUERY, because it helped summarize query mentions. The Document Feed "condenses all of the essential information and sort of leaves out all the extra stuff," said P1. Said P2, "I found [the Document Feed] useful, especially the expand button. If I click expand I can see a rundown of the mentions right after the title without seeing the article." P5 reported using the Document Feed to "do some … simple kind of topic modeling in my own head … just to see if I could pull out any … themes there." P5 added that, "having this here [i.e. Document Feed] is really helpful to kind of see what they're talking about." P3 and P4 discussed the Document Feed while describing the importance of context in historical research; we include their feedback on this feature in Section 6.4.

Several participants also reported that the **Document Viewer** helped during their research. For instance, P3 reported that automatic in-text highlighting in the feed was very helpful. "I'm a visual person. So I'm looking for the words. I like that they're in purple and green … the words that you've given me the pop out … and I can see if it's a pro or con article pretty quickly just from that." P2 said he used the Document Viewer to "provide detail."

P1, P2 and P5 noted that CLIOQUERY's linked **Document Feed and Document Viewer** served complementary purposes. They described how the Document Feed provided a summary of the query term, while the Document Viewer provided necessary and complementary details. "You need both [the Document Feed and Viewer]," said P2. "With just the Document Feed I won't be able to get the full picture of the story. And with just the Document Viewer I will not be able to trace the mentions quite comprehensively and specifically." P2 then added, "as a researcher, it's important to see things in detail. If you just conclude from what you see in the Document Feed you are not going to get an objective picture of the context of the story line. But if you see the Document Feed, see the mentions, see what they imply, and then you want to understand the context of the story you are going to get to the Document Viewer." P1 said, "I like having both the Document Feed and the Document Viewer, side by side. [The Document Viewer helps with] reading for more depth when I want more depth and [the Document Feed] helps with ... quick scans pretty easy." Similarly, P5 explained, "I see [the Document Feed and Viewer] working together really well ... I start by looking at the feed to kind of pick out the articles that would want to kind of dive into deeper and then I go into the Document Viewer."

P4 and P5 specifically mentioned that complementary linked views from the Document Feed and Document Viewer helped with **mention gathering and analysis**, as compared to a baseline keyword document search system. "A lot of a lot of databases that we work with, do something similar to this [i.e. the Document Feed]," said P5, while describing a search engine results page. "But you often then have to click on the article to go into the article to get to that reading … here it is nice that it was just kind of next to it and you can scroll through it." Similarly, P4 described the difficulties of context switching between documents from the Google search engine results page. "Obviously, it's is a time saver" she said, comparing CLIOQUERY to the keyword document search system. "You can tell … just using the editorials at one newspaper."

Two participants relied on CLIOQUERY'S **filtering system** to investigate their research topics. P1 investigated the *NYT* editorial board's discussion of the query term "Falluja" using the filter-by-subquery feature (e.g. searching for "Falluja" and "resistance" or "Falluja" and "terrorist"). "It's pretty interesting to me that I get three hits with the words Falluja and resistance and only one with the word terrorist," he said. "That would suggest a certain orientation from the editorial board that will

ACM Article No:26 Anon.

be unexpected." P2 found the filter-by-count feature very helpful. "Oh, this is good," he said, while testing out the slider. "It gets us through to the most important, the most critical pieces that we want to read."

6.3 Some disavow obligation to perform comprehensive review, noting high costs

 During needfinding, interviewees emphasized the importance of comprehensively reviewing all available evidence. However, to our surprise, during the expert interview study, P4 explicitly disavowed an obligation to search comprehensively. "I don't feel like I have an obligation to look at everything," she said. "I have an obligation to get an overview and I think you know, with a completely unscientific measure of, oh, I think I've got enough now." Similarly, P1 commented that, "I don't think anyone actually does it [search comprehensively]." He went on "A lot of people pretend they do it … [but] in terms of like visiting archives … everyone's skimming … they already know what they're looking for and they're just trying to find it." P2 pointed out that comprehensive manual review was desirable but ultimately had high costs. "I am not saying we should get rid of personal scrutiny, the way you do it yourself. [But] you want to save time. If you do it [i.e. read] one-by-one it wastes too much time." We discuss ambiguity surrounding comprehensive review in Section 9.

6.4 Context is crucial in historical research, so some are wary of text summarization

Like during needfinding (Section 3.3.3), participants often emphasized the importance of context in historical research. For instance, P3 described extensive research to prepare for oral history interviews in order to "get that context to be able to ask them the questions that I asked them." P2 also reported that context is "very important" for historians, as it "helps you understand why things are what they are."

Some users' emphasis on context informed their feedback on the Document Feed. While P1, P2 and P5 found the Document Feed useful (Section 6.2), P3 and P4 expressed reservations, because they felt they needed more context to reach conclusions. P3 took the more extreme position. "For me, I don't know if [the Document Feed] is necessary," she said. "As a history scholar, you can't take things out of context. You need to know the bigger context." On the other hand, P4 reported that she would need more context (i.e. longer extractions from news stories) before the feature would be useful. "The more context I can take in within as compact a time frame and compact a format, but sufficiently informative [the better]" she said. "But I think these [shortened sentences in the Document Feed] might have to be longer for that to work."

6.5 Some users recognize a tradeoff between neutral review and limited time

During needfinding and prototyping, interviewees often stressed the importance of avoiding possible bias from software in historical research. But during our expert interview study, P4 reported that she relied on black-box relevance models to direct her attention while searching archives. "I do try to use the chronological sorting [when using ProQuest]," said P4. "But it is ... too much to wade through. If your corpus is reasonably big then you have to have a relevance kind of algorithm in there. Otherwise, it's just going to be too frustrating." P4 also recognized that reliance on ranking introduces confounds. "I think it would be appropriate to make people look at all of the irrelevant stuff," she said. "So they realize the algorithm is pulling the relevant stuff for you ... but you can't make the search s*** for people just to sort of make that point."

On the other hand, P5 liked how CLIOQUERY used filtering to avoid potential bias. "I think it's better that its just showing everything," he explained. "I prefer having everything there to kind of whittle down ... as opposed to having certain things like cherry-picked ... I guess it's never super clear to me why certain things might be moved to the top of results ... it raises questions about how things are ordered and how they're brought to light."

As I3 predicted (Section 3.3.4), P1 described relying on the search function of the New York Times website [95], without understanding how the site was ranking search results by relevance. "I wasn't super aware of how they were pulling up articles for me ... They rank it in terms of views right?" he said. He added, "I just don't, you know, have the knowledge of how to navigate these ... search engines well enough." We discuss heterogenous feedback on algorithmic bias in history in Section 9.4.

6.6 Access, integrity and integration are important to current practices

Many participants commented on the importance of access, integrity and integration in describing their current practices with newspaper archives (see also Section 10). P1 reported gathering news articles on U.S.-Iraqi relations from around the web "for years" by using search engines like Google or the *New York Times* website [95], saving these articles to the Internet Archive [25] and then organizing this collection using the software program Omeka [27]. This participant pointed out that CLIOQUERY "assumes you have found all the stuff you want to work with," which is not true for his current research. P2 said that he had to rely on physical archives of print newspapers in Zimbabwe, which required burdensome international travel. P3 said that she rarely used newspapers in her own research, because many newspaper archives are often inaccessible behind paywalls, and P4 emphasized the need for better optical character recognition technology to improve search over printed newspapers. P5 reported that he "used Zotero a lot" to store and organize archival sources; he liked that Zotero is open-source and integrates with Microsoft Word.

7 FIELD STUDY

 In their review of design study methodology, Sedlmair *et al.* emphasize the importance of deploying a designed solution "in the wild" to test if new software helps "real users" solve "real problems" with "real data" [115]. Thus, we deployed CLIOQUERY over the web in a field study for two historians, who used the tool to answer questions from their own research. Unlike in the expert interview study, during the field study, historians investigated questions over multiple meetings, and tried to reach substantive rather than preliminary conclusions. We believe that this evaluation offers more realistic but also less uniform feedback than the one-hour expert interviews described in Section 5.

7.1 Procedure

We recruited two historians, H1 and H2 through convenience sampling [47]; the Appendix includes details on their backgrounds. H1 and H2 did not participate in the initial design or development of CLIOQUERY, to avoid what Sedlmair et al. [115] describe as a potential source of bias. During the field study, one member of our research team conducted three one-on-one meetings with each historian over Zoom video chat. The first meeting was 30 minutes long and the subsequent meetings were 60 to 70 minutes long, with 1 to 3 weeks between each meeting. Each meeting in the three meeting sequence had a distinct focus. During the first meeting, the researcher presented a tutorial of the software, described the field study process, and invited the historian to describe a question related to their research. After the first meeting, a member of our research team gathered the data needed to answer the historian's research question and loaded it into CLIOQUERY (the Appendix describes this data gathering). During the second meeting, each historian learned to use the CLIOQUERY software, and performed a preliminary exploration of the data. Then, during the final meeting, each historian investigated some specific query by analyzing the comprehensive set of all mentions using the Document Feed and Document Viewer. During each meeting, the researcher observed each historian and invited the historian to think aloud [97] as they used the system. The researcher also asked the historian to describe their findings, and explain how CLIOQUERY helped or did not help answer their research question.

ACM Article No:28 Anon.

7.2 CLIOQUERY helps experts investigate by skimming, an advantage over baselines

 During the field study, H1 and H2 each used CLIOQUERY to reach substantive historical conclusions, offering additional evidence for our hypothesis (Section 4) that CLIOQUERY can help experts answer research questions from news archives.

H1 used CLIOQUERY to verify a well-known claim from Herman and Chomsky, who argue that for-profit news organizations in the United States shape public opinion towards the interests of political and economic elites [59]. To offer evidence for this theory, in their work, Herman and Chomsky assert that *The New York Times* wrote five articles in February and March of 1984 describing the Salvadoran army as a protector of El Salvador's election. To verify this result, H1 searched a *New York Times* corpus (see Appendix) for the query "election" and then used the filter-by-date feature to select articles from February and March of 1984. H1 then used the filter-by-subquery feature to identify those query results which contained the subquery "army" and then systematically reviewed all 32 matching documents, through what H1 described as "skimming highlighted parts" in the Document Viewer. By using CLIOQUERY in this manner, H1 said that they were "able to find what might be the five articles" Herman and Chomsky used to partially support their conclusions. H1 explained, "The tool is great for exactly this."

H1 found CLIOQUERY'S in-text highlighting helpful for their research task, drawing a comparison with a baseline keyword document search system (Section 2.2.1). "I like how you have the bold highlighted and colored words in the text itself," they said. "That is the advantage that this interface has over the New York Times website." H1 also explained how such highlighting reduced reading burden, compared to keyword document search, which allowed him to skim relevant documents. "What I need to know is the army described as a protector of the election [in an article]," he said. "I don't need to read every word of the article to find that out. I can look at the paragraphs where they are describing the army and I see what they are saying in those paragraphs. That is pretty useful."

H2 chose to use CLIOQUERY to study how the United States media represented female astronauts Svetlana Savitskaya and Sally Ride in the early 1980s, to gather evidence for an upcoming book about gender and space exploration. To investigate, H2 used CLIOQUERY's Document Feed and Document Viewer to review portrayal of Sally Ride in The New York Times. H2 queried for the word "Ride" and then scrolled through the Document Feed to skim over mentions of Ride in the 63 matching documents, sometimes also clicking to open individual news stories in the Document Viewer. "I have some hypotheses that I was able to develop very quickly through the experience of using this [system]," H2 reported. "One is that Ride was presented to the American public [in The New York Times] ... first as a woman and second as a scientist". H2 asked us to continue to provide access after the study, so she could continue researching her book using the tool.

CLIOQUERY'S Document Feed was particularly helpful for H2, who found that query-focused summarization offered an advantage over a baseline keyword document search system. Ride was a PhD astrophysicist turned astronaut, and H2 wanted to understand how the media portrayed her scientific credentials. The Document Feed helped H2 quickly review this information. "[Here] she's called a flight engineer," H2 said, pointing to the Document Feed. "I can see this already [without opening the document]." H2 then scrolled through the Document Feed to find shortened sentences where Sally Ride was described with her academic title (Dr. Ride), and sentences where Ride was described (or not described) as a physicist. H2 explained that she could identify this information "just doing the quick scan [in the Document Feed]". She went on to explain how she would normally research this question with The New York Times archive by opening and reading individual news stories using a web browser. "The question is," she said, "what can I do here [with CLIOQUERY] that I can't do there [i.e. on The New York Times website]?" H2 continued, "It's exploring the left hand Document Feed here. This is awesome … I am liking these short contextual pieces [i.e. shortened

sentences]". We illustrate this comparison in Figure 2; by using CLIOQUERY, H2 was able to easily gather and analyze mentions of Ride across the corpus.

8 A QUANTITATIVE COMPARISON WITH KEYWORD DOCUMENT SEARCH TOOLS

8.1 A crowdsourced historical reading comprehension task

 We designed a crowdsourced historical reading comprehension task to compare CLIOQUERY with a keyword document search system (IR), which we consider to be a baseline tool for historical mention gathering and analysis (Section 1 and 2). Our task is modeled after a real historical question from P2 (Section 5), who used CLIOQUERY to investigate if *The New York Times* portrayed the controversial figure Robert Mugabe as a corrupt authoritarian, or as a hero of Zimbabwe's fight for independence.

In our crowdsourced adaption of P2's research task, we presented crowdworkers with one of two text analytics tools loaded with the exact same small corpus of 12 *New York Times* editorials mentioning Robert Mugabe, published from January, 2001 to June, 2003. We then asked participants find and remember everything the *The New York Times* wrote about Robert Mugabe using their tool. Because historians have only so much time for a given research project (Section 3.3.2), we limited participants to exactly six minutes to complete the task. After six minutes, we presented nine reading comprehension questions about *New York Times* coverage of Mugabe, and observed the total number of correct answers for each participant.

To ensure that our reading comprehension task was as objective and neutral as possible, we created reading questions using the Wikipedia page for Robert Mugabe [137]. Specifically, we used a semi-automated procedure based on tf-idf sentence vectors (described in detail in the Appendix) to identify Mugabe facts from Wikipedia reported in *New York Times* editorials about Mugabe. We then selected four facts from Wikipedia reported in the 12 editorials in the corpus, and five facts from Wikipedia that were not reported in the 12 editorials. These five facts were reported in some other *New York Times* editorial that was not presented to participants (because the editorial was published before or after January, 2001 to June, 2003). This process created a list of nine total Mugabe facts from Wikipedia.

To evaluate reading comprehension, we presented all nine facts in randomized order, and asked participants to select those facts which appeared in the articles they had reviewed during the task. To get a perfect score of nine out of nine correct answers without guessing, ¹⁹ a participant would have to find and remember the four Mugabe facts reported in the editorials shown during the task, without selecting any of five facts that were not reported in editorials. The Appendix includes a screenshot showing the questions, as presented to participants.

8.2 Experiment design and experiment details

We compared CLIOQUERY with a keyword document search system using a between-subjects experiment design, with U.S. masters workers recruited via Amazon Mechanical Turk and paid \$3 to complete the 20-minute experiment. (Amazon confers the master designation on crowd workers with a record of success in crowd tasks.) In the experiment, participants were randomly assigned to complete the reading comprehension task, using either CLIOQUERY or a baseline IR keyword document search interface. We then measured the difference in the mean number of total correct answers from workers in each group, to determine if CLIOQUERY helped people find and remember information about Robert Mugabe, as compared to a baseline.

We implemented the IR system using Whoosh, an open-source Python keyword document search tool which ranks results using the common BM-25 metric. The Appendix contains a

 $^{^{19}}$ In this task, a participant would have a $.5^9 * 100 = 0.195\%$ chance of correctly guessing all 9 answers.

ACM Article No:30 Anon.

screenshot of this baseline interface. Whoosh²⁰ is similar to other traditional keyword document search tools like Lucene. We implemented the IR system using Whoosh because it is not possible to compare ClioQuery to proprietary black-box systems like *New York Times* search [95] or Google (which may return different results depending on geography and user background).

To ensure fair comparison, we tuned Whoosh to be most similar to CLIOQUERY. Specifically, Whoosh accepts a number of configuration parameters which govern how the system creates snippets on the results page (see Section 2.2). Because such snippets are similar to the snippets in the CLIOQUERY Document Feed, we adjusted the Whoosh snippet parameters so that that the IR snippets were as close as possible in length to the shortened sentences in the CLIOQUERY Document Feed. Further details on the tuning procedure are described in the Appendix. We also adjusted the IR system to use the same font size as CLIOQUERY.

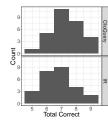
To minimize possible variation in worker behavior, we hard-coded the CLIOQUERY system and the IR system to use the query "Mugabe" during the experiment. To rank the 12 documents in the corpus using the IR system, we loaded the IR tool with all *New York Times* editorials published between 1987 and 2007 that include the word "Zimbabwe," and assigned a date filter to select only those results which were published from January, 2001 to June, 2003. We similarly used CLIOQUERY's filter-by-date control to filter select the same exact 12 documents from the same *New York Times* corpus [110]. We also removed CLIOQUERY interface elements which are not relevant for the task, such as the system guide, corpus selection control and filter-by-subquery feature.

8.2.1 Experiment sequence and experiment pretest. At the start of the experiment, participants in each condition watched a roughly one minute training video describing how to use their randomly assigned interface. CLIOQUERY participants also then completed two brief training screens (shown in supplemental materials) to gain practice using the CLIOQUERY slider and CLIOQUERY expand button. Following this training, participants in each condition then used their interface to complete a three minute pretest using a small corpus of six New York Times editorials mentioning "Iraq". The pretest was very similar to the main task. In the pretest, each interface was hard-coded to use the query "Falluja," and participants were instructed to find and remember everything the New York Times wrote about Falluja using their tool. After 3 minutes, participants were then presented with a screen with four facts about U.S. involvement in Falluja (included in supplemental materials), and asked to identify which facts were reported in the six articles. Because only one fact from the list was reported in the articles, to get a perfect score of 4 out of 4 on the pretest, workers had to both correctly identify the reported fact, and refrain from guessing any of the other three facts.

Because the Falluja pretest was very easy, we believe that participants who were not able to complete it perfectly were not participating in good faith, and we screened out data from these suspected careless workers in computing results. (Careless workers were still permitted to complete the main task, and were paid for their work). Additionally, throughout the pretest and training, we logged each time participants made a mistake on task instructions (e.g. trying to skip the training video without watching), and we screened out data from these workers in computing results. In their highly-cited on crowdsourcing for HCI, Kittur et al. [69] emphasize the importance of detecting suspect responses in crowdsourced settings, and mention the utility of qualifying users via automated pretests on the Amazon Turk platform. We defined filtering procedures before data collection, to mitigate potential data mining biases. Nevertheless, for maximal transparency, we also report results on unfiltered data, which we believe to have greater noise.

8.2.2 Results. [Abe: in the table, list pairs of significant results]

²⁰ https://whoosh.readthedocs.io/



SD	Mean	condition
(1.04) (1.11)	7.310345 6.769231	ClioQuery IR
	6.769231	IR ,

Fig. 7. A figure and a table, side-by-side

[Abe: I am imagining we measure the recall of evidence from Wikipedia. Seeing the sentences on screen is necessary but not sufficient to comprehending the information they contain.]
[Abe: can we justify our T/F questions with AP history tests? Investigate.]

9 DISCUSSION

9.1 CLIOQUERY suggests new features and directions for interactive text analysis

Much prior work in interactive text analysis focuses on helping people investigate bodies of documents, by presenting textual units like topics [72], events [76] or thematic hierarchies [12]. But motivated by the needs of historians and archivists, CLIOQUERY instead proposes and tests a new approach to interactive corpus investigation, organized around the analysis of query mentions in context (see Section 2.3).

To help people investigate this atypical "unit of analysis" [21], CLIOQUERY employs new text summarization techniques from the NLP literature to create a skimmable summary of a query term across a corpus. The system then presents such skimmable summaries alongside more traditional features from the text analytics literature, such as linked views and in-text highlighting, to help people easily and transparently review summary text in underlying documents. During expert interview and field study evaluations, many historians said that they found such features helpful for archival research. They reported skimming over query mentions in the Document Feed to gain a sense of a query's use across a corpus, and then reading highlighted mentions in the Document Viewer for more context and detail. Several specifically mentioned that these components helped with mention gathering and analysis.

This query-oriented approach suggests new directions for interactive text analytics, in other query-oriented settings. For instance, some marketing applications identify salient words and phrases in online forums [48, Section 4.1]; CLIOQUERY'S query-focused summaries and linked views might help marketing analysts using such systems understand what people say about products online. Such features might also be applied in existing text analytics systems. For instance, people might formulate a query using overview-oriented features such as word clusters, and then investigate this query word using a CLIOQUERY-style Document Feed and Document Viewer.

9.2 CLIOQUERY tests an idea: "Text and its affordances should be taken seriously"

Researchers have proposed many approaches to text visualization, which map high-dimensional text to two-dimensional graphical representations like time series plots (e.g. ThemeRiver [56]) or bubble diagrams (e.g. EventRiver [76]). By contrast, CLIOQUERY's Document Feed and Document Viewer do not map text data to a graphical representation. Instead, CLIOQUERY uses text summarization methods from NLP to extract and present spans of text from a corpus for people to read, using automatic in-text highlighting to facilitate skimming. In this sense, CLIOQUERY follows the advice of Sultanum et al. [126], who suggest that "text and its affordances should be taken seriously"

ACM Article No:32 Anon.

 in text analytics, by making text itself "a central piece of the visualization." Viewed through the lens of this recommendation, CLIOQUERY reflects one strategy for a text analytics system fundamentally organized around displaying spans from a corpus (instead of mapping text data to some other graphical form). Note that other work in text analytics also emphasizes the importance of showing underlying documents. For instance, creators of the Jigsaw system write that "interactive visualization cannot replace the reading of reports" [49], and creators of both Overview [124, Sec. 5] and ThemeRiver [56, Sec. 7] describe similar findings.

9.3 User feedback on summarization has implications for natural language processing

CLIOQUERY applies particular ideas from query-focused summarization in NLP for interactive text analysis. However, building and evaluating a user-facing summarization system for historians forced us to reexamine several core assumptions from the literature on text summarization. In particular, early versions of CLIOQUERY applied standard optimization-based summarization methods [86] to select "important" information from a corpus, much like prior temporally-oriented language engineering systems such as HistDiv [120], TimeMine [2] and TimeExplorer [84], which each attempt to automatically identify most-relevant information, based on a query.

However, during needfinding and prototyping, we found that some historians and archivists strongly disliked this approach. Experts reported that they needed to understand why the computer was showing particular summaries, before they could actually draw conclusions from the output (see prototypes in the Appendix). Based on this feedback, in later versions of CLIOQUERY, we stopped trying to extract "important" mentions of a query term in search results. Instead, we decided to shorten and present every single sentence mentioning a user's query in the Document Feed, and allow people to easily examine such shortenings in context in the Document Viewer. During our expert interview and field study and evaluations, we found that this approach was more successful. We hypothesize that experts liked this format, because they could understand why CLIOQUERY showed query shortenings, and thus use CLIOQUERY output in their research.

Our experiences might have implications for NLP, where research in summarization typically focuses on generating summaries which best match "gold" references [32, 93], without worrying about how to explain how summaries are formed. In particular, much recent work on abstractive summarization in NLP [60, 109] seeks to generate summary passages that do not occur in the input text. Because such abstractive output can not be checked against underlying sources, and because such methods also currently suffer from frequent factual errors [71], much more research towards trustworthy abstraction may be required, before these approaches might be applied towards social research.

9.4 Comprehensive and unbiased search costs time; transparency might help

During needfinding interviews, historians and archivists often emphasized the importance of directly and comprehensively examining all evidence relevant to a given research question, without allowing black-box algorithms to influence their conclusions. We thus designed ClioQuery to help people keep track of which query mentions they have reviewed in a corpus, without potential bias from algorithmic ranking. Yet feedback on these aspects of ClioQuery was mixed (Section 6.3 and 6.5). While some appreciated how ClioQuery used filters instead of ranking to narrow down search results, others reported that truly forgoing algorithmic curation required the researcher to spend too much time reading irrelevant documents. For instance, some admitted that they often no have choice but to trust computer models of relevance to find evidence in archives, because keyword search often turns up far more documents than they can possibly review. While historians do sometimes work with smaller corpora (Section 4.3), this issue would be particularly problematic in larger archives, where some queries will be mentioned many times.

Why did some express deep commitment to full manual review of evidence during needfinding interviews, while others admit that they had to trust search engines to select evidence during system evaluation? There seem to be at least two possibilities. One possibility is that historians and archivists might express commitment to comprehensive review when describing their ideal practices, but remember the limitations of this ideal when faced with a real task during system evaluation. (Some approaches to needfinding in HCI emphasize the limits of user interviews [10]). Another possibility is that there is simply some variation in historians' commitment to comprehensiveness. Some but not all historians may feel required to comprehensively review all evidence during research, possibly based on intellectual background or subfield. Other work in text analytics has found similar variation among doctors [127, Sec. 4.3.5].

Future researchers might resolve the contradiction between experts' stated commitments to comprehensive review and the realities of inevitable tradeoffs between recall and time [104, Fig. 6] with improved user interfaces. Specifically, systems might transparently show which documents are selected or hidden by an algorithm, and allow people to easily override and investigate any document ranking decisions from a machine. Such features would be particularly important for larger corpora, where historians would not be able to review all query mentions in context. Research on tools for visually and interactively refining search results [114] might offer a useful starting point. Features which help groups of historians to collaborate during search could also enable teams of researchers to comprehensively review evidence from larger corpora.

10 LIMITATIONS AND FUTURE WORK

Because it was difficult and expensive to recruit and interview highly-trained experts, in this study we report results from in-depth meetings with a small sample of humanists. While such one-on-one interviews provided rich feedback, the opinions of our participants likely only approximate the true requirements of all historians and archivists. In the future, we thus plan to take steps to facilitate adoption, in order to learn more about user needs. In particular, we found that historians have to collect, organize and sometimes digitize news stories, before they are ready to gather and analyze query mentions (Section 6.6). We thus plan to add features for importing news stories into ClioQuery from existing tools like Zotero and the Internet Archive. Additionally, throughout this work, we assume that query mentions are defined by exact string matches. This simplifying assumption allows us to focus on user experience and interaction, but has clear limitations. For instance, authors sometimes refer to "Reagan" using the nickname "Dutch". Automatically detecting such aliases (and other deviations from exact string matching) will be important for future work.

11 CONCLUSION

 This study describes the design and evaluation of the CLIOQUERY text analytics system. Where prior tools focus on the analysis of textual units like topics [72], events [76] or hierarchies [12] (Section 2.3), CLIOQUERY is organized around investigation of query words in context, which form the system's central "unit of analysis" [21]. CLIOQUERY's unusual emphasis on query words emerged from our study into the needs and practices of historians and archivists, who find and analyze query mentions in context in order to answer questions about society. In particular, working with and studying historians revealed that analyzing change across time, undertaking comprehensive review of evidence, evaluating contextual information and conducting neutral observation were each central to the practice of historical research. CLIOQUERY thus introduces text summarization methods from NLP to the literature on text analytics to create a novel skimmable summary of the comprehensive set of all mentions of a query term in an archive. The system then uses more traditional visual analytics features like linked views and automatic in-text highlighting to show summary text within the context of underlying news stories. Additionally, because some historians

ACM Article No:34 Anon.

expressed wariness of algorithmic bias, CLIOQUERY offers features to filter out query mentions in a neutral manner, without relying on black-box algorithms that might influence research. We tested CLIOQUERY in two separate user studies with historians, where we found that CLIOQUERY's approach to organizing and presenting query mentions could help experts answer real questions from news archives. Many historians reported that CLIOQUERY'S Document Feed facilitated rapid analysis of query mentions, and that CLIOQUERY'S linked Document Viewer offered complementary context and detail. [Abe: quant study sentence here]

Finally, our work on CLIOQUERY suggests possible new directions for interactive text analysis. In particular, CLIOQUERY's Document Feed tests new text simplification methods in a visual analytics system, in order to create novel query-oriented summaries for historical research. CLIOQUERY then links summaries with a Document Viewer, to help historians gather and analyze query mentions in context. This same combination of novel and traditional features could be applied in other query-oriented settings, where people might also need to investigate query words in context. For instance, some marketing applications suggest notable keywords from comments in online forums [48, Section 4.1]. In the future, CLIOQUERY methods might be applied to help marketers gather and analyze keyword mentions, or to help others investigate queries in other domains.

REFERENCES

1618

1619

1620

1621

1623

1624 1625

1626

1627

1628

1629

1631

1632

1633

1635

1637

1645

1649

1651

1653

1654

1655

1657

1658

1659

1661

1662

1663

1664

- [1] Eric Alexander, Joe Kohlmann, Robin Valenza, Michael Witmore, and Michael Gleicher. 2014. Serendip: Topic model-driven visual exploration of text corpora. In VAST.
- [2] Robert B. Allen and Robert Sieczkiewicz. 2010. How Historians Use Historical Newspapers. In *Proc. ASIS & T.* American Society for Information Science.
- [3] Joshua Bambrick, Minjie Xu, Andy Almonte, Igor Malioutov, Guim Perarnau, Vittorio Selo, and Iat Chong Chan. 2020. NSTM: Real-Time Query-Driven News Overview Composition at Bloomberg. In ACL: System Demonstrations.
- [4] Regina Barzilay and Kathleen R. McKeown. 2005. Sentence Fusion for Multidocument News Summarization. Computational Linguistics 31, 3 (2005).
- [5] Eric P. S. Baumer, David Mimno, Shion Guha, Emily Quan, and Geri K. Gay. 2017. Comparing grounded theory and topic modeling: Extreme divergence or unlikely convergence? *Journal of the Association for Information Science and Technology* 68, 6 (2017), 1397–1410.
- [6] Nicolas Garcia Belmonte. 2014. Extracting and visualizing insights from real-time conversations around public presentations. In IEEE VAST.
- [7] Emily M. Bender. 2013. Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax. Synthesis Lectures on Human Language Technologies 6, 3 (2013). Publisher: Morgan & Claypool Publishers.
- [8] William R. Black. 2018. How Watermelons Became Black: Emancipation and the Origins of a Racist Trope. Journal of the Civil War Era 8, 1 (2018), 64–86.
- [9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. JMLR (2003).
- [10] Jeanette Blomberg and Mark Burrell. 2012. An Ethnographic Approach to Design. In The Human-Computer Interaction Handbook, Julie A. Jacko (Ed.). CRC Press, New York.
- [11] Lauren Bradel, Chris North, Leanna House, and Scotland Leman. 2014. Multi-model semantic interaction for text analytics. In VAST.
- [12] M. Brehmer, S. Ingram, J. Stray, and T. Munzner. 2014. Overview: The Design, Adoption, and Analysis of a Visual Document Mining Tool for Investigative Journalists. IEEE TVCG 20, 12 (2014).
 - [13] Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In WWW.
 - [14] Erika Bsumek, Matthew O'Hair, Ian Diaz, and Braeden Kennedy. Accessed Dec 24, 2020. ClioVis. https://web.archive.org/web/20201224213752/https://cliovis.org/
 - [15] Andreas Buja, John Alan McDonald, John Michalak, and Werner Stuetzle. 1991. Interactive data visualization using focusing and linking. In Proceedings of the 2nd conference on Visualization. IEEE.
 - [16] Donald Owen Case. 1991. The Collection and Use of Information by Some American Historians: A Study of Motives and Methods. The Library Quarterly 61, 1 (1991), 61–82.
 - [17] Alexandra Chassanoff. 2013. Historians and the Use of Primary Source Materials in the Digital Age. *The American Archivist* 76, 2 (2013), 458–480.
 - [18] Danqi Chen and Christopher Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *EMNLP*.

[19] Daniel Chomsky and Scott Barclay. 2010. The Mass Media, Public Opinion, and Lesbian and Gay Rights. Annual Review of Law and Social Science 6, 1 (2010), 387–403.

- [20] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. 2012. Termite: Visualization Techniques for Assessing Textual Topic Models. In AVI.
- [21] Jason Chuang, Daniel Ramage, Christopher Manning, and Jeffrey Heer. 2012. Interpretation and Trust: Designing
 Model-Driven Visualizations for Text Analysis. In CHI.
- [22] Charles L. A. Clarke, Eugene Agichtein, Susan Dumais, and Ryen W. White. 2007. The Influence of Caption Features on Clickthrough Patterns in Web Search. In SIGIR.
 - [23] James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. Journal of Artificial Intelligence Research 31 (2008), 399–429.
 - [24] Chrome Contributors. 2021. Search the web on Chrome. https://web.archive.org/web/20201105231921/https://support.google.com/chrome/answer/95440
 - [25] Internet Archive Contributors. 200x. Internet Archive. Accessed Dec 8, 2020.
- [26] Lucene contributors. 2020. Apache LuceneTM 7.4.0 Documentation. https://lucene.apache.org/core/7_4_0/index.html
 Accessed Jun 15, 2020.
 - [27] Omeka Contributors. 200x. Omeka. Accessed Dec 15, 2020.

1674

1675

1676

1680

1681

1684

1692

1694

1695

1698

1700

1701

1705

1711

1712

1713 1714 1715

- [28] W Bruce Croft, Donald Metzler, and Trevor Strohman. 2015. Search engines: Information retrieval in practice. Pearson Education.
- [29] Weiwei Cui, Shixia Liu, Li Tan, Conglei Shi, Yangqiu Song, Zekai Gao, Huamin Qu, and Xin Tong. 2011. TextFlow:
 Towards Better Understanding of Evolving Topics in Text. IEEE TVCG 17, 12 (2011).
 - [30] Weiwei Cui, Shixia Liu, Zhuofeng Wu, and Hao Wei. 2014. How Hierarchical Topics Evolve in Large Text Corpora. *IEEE TVCG* 20, 12 (2014).
 - [31] Margaret Stieg Dalton and Laurie Charnigo. 2004. Historians and Their Information Sources. *College & Research Libraries* 65, 5 (2004).
- [687 [32] Dipanjan Das and André FT Martins. 2007. A survey on automatic text summarization. Technical Report. Carnegie Mellon University.
 - [33] N. Diakopoulos, M. Naaman, and F. Kivran-Swaine. 2010. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In IEEE VAST.
 - [34] Anthony Don, Elena Zheleva, Machon Gregory, Sureyya Tarkan, Loretta Auvil, Tanya Clement, Ben Shneiderman, and Catherine Plaisant. 2007. Discovering Interesting Usage Patterns in Text Collections: Integrating Text Mining with Visualization. In *CIKM* (Lisbon, Portugal). 10.
 - [35] Wendy Duff, Barbara Craig, and Joan Cherry. 2004. Historians' Use of Archival Sources: Promises and Pitfalls of the Digital Age. The Public Historian 26, 2 (2004), 7–22.
 - [36] Wendy M. Duff and Catherine A. Johnson. 2002. Accidentally Found on Purpose: Information-Seeking Behavior of Historians in Archives. The Library Quarterly 72, 4 (2002), 472–496.
- [37] M. Dörk, S. Carpendale, C. Collins, and C. Williamson. 2008. VisGets: Coordinated Visualizations for Web-based
 Information Exploration and Discovery. *IEEE TVCG* 14, 6 (2008).
 - [38] Marian Dörk, Nathalie Henry Riche, Gonzalo Ramos, and Susan Dumais. 2012. PivotPaths: Strolling through Faceted Information Spaces. TVCG 18, 12 (2012).
 - [39] Maud Ehrmann, Estelle Bunout, and Marten Düring. 2019. Historical Newspaper User Interfaces: A Review. In International Federation of Library Associations World Library Information Conference.
 - [40] Jacob Eisenstein. 2019. Introduction to Natural Language Processing. MIT Press, Cambridge, MA.
- 1702 [41] Henry Feild, Ryen W. White, and Xin Fu. 2013. Supporting Orientation during Search Result Examination. In CHI.
- [42] Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence Compression by Deletion with LSTMs. In EMNLP.
 - [43] Katja Filippova and Yasemin Altun. 2013. Overcoming the Lack of Parallel Data in Sentence Compression. In EMNLP.
- [44] Katja Filippova and Michael Strube. 2008. Dependency Tree Based Sentence Compression. In *Proceedings of the Fifth*International Natural Language Generation Conference.
- [45] Eric Foner. 1995. Free soil, free labor, free men: The ideology of the Republican Party before the Civil War. Oxford
 University Press, New York.
- [46] William Gaver. 2014. Science and Design: The Implications of Different Forms of Accountability. In Ways of Knowing in HCI, Judith S. Olson and Wendy A. Kellogg (Eds.). Springer, New York, NY.
 - [47] Lisa Given and Kristie Saumure. 2008. Convenience Sample. In *The SAGE Encyclopedia of Qualitative Research Methods*, Lisa Given (Ed.). SAGE Publications, Inc., Thousand Oaks, California.
 - [48] Natalie Glance, Matthew Hurst, Kamal Nigam, Matthew Siegler, Robert Stockton, and Takashi Tomokiyo. 2005. Deriving Marketing Intelligence from Online Discussion. In KDD.

ACM Trans. Interact. Intell. Syst., Vol. xx, No. xx, Article ACM Article No. Publication date: January 2021.

ACM Article No:36 Anon.

1716 [49] Carsten Görg, Zhicheng Liu, and John Stasko. 2013. Reflections on the evolution of the Jigsaw visual analytics system.

1717 Information Visualization (2013).

- 1718 [50] John D. Gould and Clayton Lewis. 1985. Designing for Usability: Key Principles and What Designers Think. *Commun.*1719 ACM 28, 3 (1985), 300–311.
- [51] Shawn Graham, Scott Weingart, and Ian Milligan. 2012. Getting started with topic modeling and MALLET. https://programminghistorian.org/en/lessons/topic-modeling-and-mallet. Accessed: 2021-06-02.
- [52] Abram Handler, Su Lin Blodgett, and Brendan T. O'Connor. 2016. Visualizing textual models with in-text and word-as-pixel highlighting. *ICML Workshop on Human Interpretability in Machine Learning* (2016).
- [53] Abram Handler, Brian Dillon, and Brendan T. O'Connor. 2019. Human acceptability judgements for extractive sentence compression. *ArXiv* (2019). https://arxiv.org/pdf/1902.00489
 - [54] Abram Handler and Brendan O'Connor. 2018. Relational Summarization for Corpus Analysis. In NAACL.

1725

1728 1729

1730

1733

1735

1737

1739

1742

1743

1749

1750

1751

1759

- [55] Abram Handler and Brendan O'Connor. 2019. Ouery-focused Sentence Compression in Linear Time. In EMNLP.
- 1726 [56] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. 2002. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE TVCG* 8, 1 (Jan 2002), 9–20.
 - [57] Marti A. Hearst. 1995. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In CHI.
 - [58] Jeffrey Heer and Ben Shneiderman. 2012. Interactive Dynamics for Visual Analysis. *Commun. ACM* 55, 4 (2012), 45–54.
- [59] Edward S. Herman and Noam Chomsky. 1988. Manufacturing consent: the political economy of the mass media.
 Pantheon Books, New York, NY.
 - [60] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In NIPS.
 - [61] O. Hoeber and Xue Dong Yang. 2006. The Visual Exploration of Web Search Results Using HotMap. In Tenth International Conference on Information Visualisation (IV'06).
 - [62] Jennifer Hoewe and Geri Alumit Zeldes. 2012. Overturning Anti-Miscegenation Laws: News Media Coverage of the Lovings' Legal Case Against the State of Virginia. Journal of Black Studies 43, 4 (2012), 427–443.
 - [63] Enamul Hoque and Giuseppe Carenini. 2016. Interactive Topic Modeling for Exploring Asynchronous Online Conversations: Design and Evaluation of ConVisIT. *TiiS* 6, 1, Article 7 (2016).
 - [64] Yuening Hu, Jordan Boyd-Graber, and Brianna Satinoff. 2011. Interactive Topic Modeling. In ACL.
- [65] E. Isaacs, K. Damico, S. Ahern, E. Bart, and M. Singhal. 2014. Footprints: A Visual Search Tool that Supports Discovery
 and Coverage Tracking. *IEEE TVCG* 20, 12 (2014), 1793–1802.
 - [66] Mahmood Jasim, Pooya Khaloo, Somin Wadhwa, Amy X. Zhang, Ali Sarvghad, and Narges Mahyar. 2020. CommunityClick: Towards Improving Inclusivity in Town Halls. In CSCW.
 - [67] Hannah Kim, Barry Drake, Alex Endert, and Haesun Park. 2021. ArchiText: Interactive Hierarchical Topic Modeling. TVCG 27, 9 (2021).
- [68] Minjeong Kim, Kyeongpil Kang, Deokgun Park, Jaegul Choo, and Niklas Elmqvist. 2017. TVCG. IEEE Transactions on
 Visualization and Computer Graphics 23, 1 (2017).
- 1747 [69] Aniket Kittur, Ed H. Chi, and Bongwon Suh. 2008. Crowdsourcing User Studies with Mechanical Turk. In CHI.
- [70] Kevin Knight and Daniel Marcu. 2000. Statistics-Based Summarization Step One: Sentence Compression. In AAAI.
 - [71] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural Text Summarization: A Critical Evaluation. In EMNLP.
 - [72] Shixia Liu, Michelle X. Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. 2012. TIARA: Interactive, Topic-Based Visual Text Summarization and Analysis. *ACM TIST*. 3, 2, Article 25 (2012).
- 1752 [73] Zhicheng Liu and Jeffrey Heer. 2014. The Effects of Interactive Latency on Exploratory Visual Analysis. *IEEE TVCG* (2014).
- 1754 [74] H. P. Luhn. 1960. Key word-in-context index for technical literature (kwic index). *American Documentation* 11, 4 (1960), 288–295.
- [75] Jack Lule. 2002. Myth and Terror on the Editorial Page: The New York Times Responds to September 11, 2001.
 Journalism & Mass Communication Quarterly 79, 2 (2002), 275–293.
- 1757 [76] D. Luo, J. Yang, M. Krstajic, W. Ribarsky, and D. Keim. 2012. EventRiver: Visually Exploring Text Collections with
 1758 Temporal References. *IEEE TVCG* 18, 1 (Jan 2012), 93–105.
 - [77] Saturnino Luz and Shane Sheehan. 2014. A Graph Based Abstraction of Textual Concordances and Two Renderings for Their Interactive Visualisation. In AVI.
- [78] Virpi Lyytikäinen, Pasi Tiitinen, and Airi Salminen. 2000. Graphical Information Models as Interfaces for Web
 Document Repositories. In AVI.
- [79] Trent MacNamara. 2014. Why "Race Suicide"? Cultural Factors in U.S. Fertility Decline, 1903–1908. The Journal of
 Interdisciplinary History 44, 4 (2014), 475–508.

[80] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval.
 Cambridge University Press, USA.

[81] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014.
The Stanford CoreNLP Natural Language Processing Toolkit. In ACL: System Demonstrations.

1768

1776

1777

1780

1791

1792

1794

1796

1798

1799

1800

1808

- [82] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. 2011. Twitinfo: Aggregating and Visualizing Microblogs for Event Exploration. In CHI.
- 1770 [83] Howard Markel, Harvey B. Lipman, J. Alexander Navarro, Alexandra Sloan, Joseph R. Michalsen, Alexandra Minna 1771 Stern, and Martin S. Cetron. 2007. Nonpharmaceutical Interventions Implemented by U.S. Cities During the 1918-1919 1772 Influenza Pandemic. JAMA 298, 6 (2007), 644-654.
- [84] Michael Matthews, Pancho Tolchinsky, Roi Blanco, Jordi Atserias, Peter Mika, and Hugo Zaragoza. 2010. Searching through time in the New York Times. In *HCIR*.
- [85] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-Rater Reliability in Qualitative
 Research: Norms and Guidelines for CSCW and HCI Practice. CSCW (2019).
 - [86] Ryan McDonald. 2007. A Study of Global Inference Algorithms in Multi-Document Summarization. In ECIR.
 - [87] Kathleen McKeown and Dragomir R. Radev. 1995. Generating Summaries of Multiple News Articles. In SIGIR.
- 1778 [88] Kathleen R. McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L. Klavans, Ani Nenkova,
 Carl Sable, Barry Schiffman, and Sergey Sigelman. 2002. Tracking and Summarizing News on a Daily Basis with
 Columbia's Newsblaster. In *HLT*.
 - [89] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Holberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. 2010. Quantitative Analysis of Culture Using Millions of Digitized Books. Science (2010).
- 1782 Elegational Aldeli. 2010. Qualitative Allalysis of Culture Using Militoris of Digitized Books. Science (2010).
 [90] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. http://arxiv.org/abs/1301.3781
 - [91] Matthew B. Miles, A. M. Huberman, and Johnny Saldaña. 2014. Qualitative data analysis: A methods sourcebook. (3 ed.). SAGE.
 - [92] John W. Mohr and Petko Bogdanov. 2013. Topic models: What they are and why they matter. *Poetics* 41 (2013), 545–569.
- [93] Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. In *Mining text data*. Springer, 43–76.
- 1789 [94] Ani Nenkova and Lucy Vanderwende. 2005. *The impact of frequency on summarization*. Technical Report. Microsoft Research.
 - [95] New York Times contributors. 2020. The New York Times search page. https://www.nytimes.com/search Accessed June 12, 2020.
 - [96] Newspaper.com contributors. 2020. newspapers.com. https://www.newspapers.com [Online; accessed 15-June-2020].
 - [97] Janni Nielsen, Torkil Clemmensen, and Carsten Yssing. 2002. Getting Access to What Goes on in People's Heads? Reflections on the Think-Aloud Technique. In NordiCHI.
 - [98] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan T. McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In LREC.
 - [99] Seyednaser Nourashrafeddin, Ehsan Sherkat, Rosane Minghim, and Evangelos E. Milios. 2018. A Visual Approach for Interactive Keyterm-Based Clustering. *TiiS* 8, 1, Article 6 (2018).
 - [100] Brendan O'Connor. 2014. MiTextExplorer: Linked brushing and mutual information for exploratory text data analysis. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces.*
- [101] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu
 Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau,
 Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-Learn: Machine Learning in Python. JMLR
 12 (2011), 2825–2830.
- [102] Peter R. Petrucci and Michael Head. 2006. Hurricane Katrina's Lexical Storm: The Use of "Refugee as" A Label for
 American Citizens. Australasian Journal of American Studies 25, 2 (2006), 23–39.
- [1806 [103] Paul Pierson. 2004. Politics in time: History, institutions, and social analysis. Princeton University Press, Princeton, New Jersey.
 - [104] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In Proceedings of international conference on intelligence analysis, Vol. 5.
- [105] Proquest LLC. 2020. Proquest. https://search.proquest.com/ Accessed June 12, 2020.
- [106] Lara Putnam. 2016. The Transnational and the Text-Searchable: Digitized Sources and the Shadows They Cast. The
 American Historical Review 121, 2 (2016), 377–402.

ACM Article No:38 Anon.

[107] William Revelle. 2020. psych: Procedures for Psychological, Psychometric, and Personality Research. Northwestern University, Evanston, Illinois. R package version 2.0.9.

- [188] Geoffrey Rockwell, Stéfan G. Sinclair, Stan Ruecker, and Peter Organisciak. 2010. Ubiquitous text analysis. paj: The Journal of the Initiative for Digital Humanities, Media, and Culture 2, 1 (2010).
- [109] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *EMNLP*.
- 1819 [110] Evan Sandhaus. 2008. The New York Times Annotated Corpus. Linguistic Data Consortium LDC2008T19 (2008).
- [111] Benjamin M Schmidt. 2012. Words alone: Dismantling topic models in the humanities. *Journal of Digital Humanities* 2, 1 (2012), 49–65.
- [112] Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In *LREC*.
- [113] Carson T Schütze and Jon Sprouse. 2014. Judgment data. In Research methods in linguistics, Robert J. Podesva and
 Devyani Sharma (Eds.). Cambridge University Press.
- [114] Cecilia Di Sciascio, Vedran Sabol, and Eduardo Veas. 2017. Supporting Exploratory Search with a Visual User-Driven Approach. *TiiS* 7, 4, Article 18 (2017).
- [115] M. Sedlmair, M. Meyer, and T. Munzner. 2012. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE TVCG* 18, 12 (2012).
- [116] Ehsan Sherkat, Evangelos E. Milios, and Rosane Minghim. 2019. A Visual Analytics Approach for Interactive
 Document Clustering. TiiS 10, 1, Article 6 (2019).
- [117] Jeannie N. Shinozuka. 2013. Deadly Perils: Japanese Beetles and the Pestilential Immigrant, 1920S-1930S. *American Quarterly* 65, 4 (2013), 831–852.
- [118] Ben Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In Proceedings of the 1996 IEEE Symposium on Visual Languages.
- [119] Jaspreet Singh, Wolfgang Nejdl, and Avishek Anand. 2016. Expedition: A Time-Aware Exploratory Search System
 Designed for Scholars. In SIGIR.
- [1835 [120] Jaspreet Singh, Wolfgang Nejdl, and Avishek Anand. 2016. History by Diversity: Helping Historians Search News Archives. In CHIR.
- [121] Janine Solberg. 2012. Googling the Archive: Digital Tools and the Practice of History. Advances in the History of Rhetoric 15, 1 (2012).
- [122] John Stasko, Carsten Görg, and Zhicheng Liu. 2008. Jigsaw: Supporting Investigative Analysis through Interactive
 Visualization. Information Visualization 7, 2 (2008), 118–132.
 - [123] Robin Stewart, Gregory Scott, and Vladimir Zelevinsky. 2008. Idea Navigation: Structured Browsing for Unstructured Text. In CHI.
 - [124] Jonathan Stray. 2016. What do Journalists do with Documents?. In Computation+Journalism Symposium.

1840

1841

- [125] Hendrik Strobelt, Daniela Oelke, Christian Rohrdantz, Andreas Stoffel, Daniel A. Keim, and Oliver Deussen. 2009.

 Document Cards: A Top Trumps Visualization for Documents. TVCG 15, 6 (2009).
- [126] Nicole Sultanum, Michael Brudno, Daniel Wigdor, and Fanny Chevalier. 2018. More Text Please! Understanding and
 Supporting the Use of Visualization for Clinical Text Overview. In CHI.
- [127] Nicole Sultanum, Devin Singh, Michael Brudno, and Fanny Chevalier. 2019. Doccurate: A Curation-Based Approach for Clinical Text Visualization. *TVCG* 25, 1 (2019).
- [128] Jenifer Tidwell. 1999. A Pattern Language for Human-Computer Interface Design. https://web.archive.org/web/20201212151556/http://www.mit.edu/~jtidwell/common_ground.html [Online; accessed 16-Jan-2021].
- 1849 [129] Jenifer Tidwell, Charles Brewer, and Aynne Valencia. 2020. Designing Interfaces (3rd ed.). O'Reilly Media, Inc.
- [130] Anastasios Tombros and Mark Sanderson. 1998. Advantages of Query Biased Summaries in Information Retrieval. InSIGIR.
- 1852 [131] Ted Underwood. 2014. Theorizing Research Practices We Forgot to Theorize Twenty Years Ago. Representations 127, 1 (2014), 64–72.
- [132] Frank van Ham, Martin Wattenberg, and Fernanda B. Viegas. 2009. Mapping Text with Phrase Nets. TVCG 15, 6
- [133] Fernanda B. Viégas, Scott Golder, and Judith Donath. 2006. Visualizing Email Content: Portraying Relationships from Conversational Histories. In *CHI*.
- 1856
 [134] Romain Vuillemot, Tanya Clement, Catherine Plaisant, and Amit Kumar. 2009. What's being said near "Martha"?
 Exploring name entities in literary text collections. In VAST.
- [135] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. 2000. Guidelines for Using Multiple Views in
 Information Visualization. In AVI.
- [136] 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.

1863 [137] Wikipedia contributors. 2021. Robert Mugabe. https://en.wikipedia.org/wiki/Robert_Mugabe [Online; accessed 1864 14-Oct-2021].

- [138] Polle T. Zellweger, Bay-Wei Chang, and Jock D. Mackinlay. 1998. Fluid Links for Informed and Incremental Link Transitions. In Hypertext.
- [139] George K. Zipf. 1949. Human Behavior and the Principle of Least Effort. Addison-Wesley.