CH-Bench: a User-oriented Benchmark for Systems for Efficient Distant Reading—Design, Performance, and Insights

Jens Willkomm[1*], Markus Raster[1], Martin Schäler[2†] and Klemens Böhm[1]

[1*]Karlsruhe Institute of Technology (KIT), Germany.
[2]University of Salzburg, Austria.

*Corresponding author(s). E-mail(s): jens.willkomm@kit.edu;
Contributing authors: markus.raster@googlemail.com; martin.schaeler@sbg.ac.at;
klemens.boehm@kit.edu;
†Work originated while author was at KIT.

Abstract

Data Science deals with the discovery of information from large volumes of data. The data studied by scientists in the humanities include large textual corpora. An important objective is to study the ideas and expectations of a society regarding specific concepts, like "freedom" or "democracy", both for today's society and even more for societies of the past. Studying the meaning of words using large corpora requires efficient systems for text analysis, so-called distant reading systems. Making such systems efficient calls for a specification of the necessary functionality and clear expectations regarding typical work loads. But this currently is unclear, and there is no benchmark to evaluate distant reading systems. In this article, we propose such a benchmark, with the following innovations: As a first step, we collect and structure various information needs of the target users. We then formalize the notion of word context to facilitate the analysis of specific concepts. Using this notion, we formulate queries in line with the information needs of users. Finally, based on this, we propose concrete benchmark queries. To demonstrate the benefit of our benchmark, we conduct an evaluation, with two objectives. First, we aim at insights regarding the content of different corpora, i.e., whether and how their size and nature (e.g., popular and broad literature or specific expert literature) affect results. Second, we benchmark different data management technologies. This has allowed us to identify performance bottlenecks.

Keywords: benchmark design, text corpus, distant reading, query performance, corpus insights

1 Introduction

Data Science deals with the discovery of new insights from large volumes of data. One important kind of such data is digital libraries or derivations of it whose content is time-stamped. A well-known example is the Google Books Ngram data set. It summarizes the Google Books corpus, which contains a large share of all books ever published [24]. For the first time ever, this lets scholars discover and access relevant information in the world's literature—if technical systems support the respective functionality and provide acceptable performance and scalability. If so, this will revolutionize scientific methods in the humanities.

Using the support of technical systems to examine large amounts of text is known as *distant reading* [31]. In a long-term cooperation between philosophers and computer scientists, we work

toward building systems which support studying hypotheses on *conceptual history*, i.e., the lexicography that studies the history of words. Conceptual history is a suitable candidate for distant reading systems since it relies only on facts in the text and not on additional interpretations [20]. Having said this, it is currently unclear which functionality needs to be supported. To this end, we have surveyed various information needs, i.e., which information conceptual historians seek. Our study reveals that one key feature is to analyze the context of words by looking for *collocations*, which ones exist, when they change and how. Collocations are words that are frequently adjacent to each other [15]. For example, some collocations for "coffee" are "drink", "hot", and "tea". Conceptual historians use this information to derive the meaning of a word, when the meaning changed, and how it changed.

Specifying a distant reading system supporting studies on conceptual history is difficult for the following reasons. Firstly, conceptual history does not have any rigorous, formalized approach how to analyze words and their contexts. Most of the previous investigations in conceptual history follow best practices, which are implicit. Secondly, it is difficult to structure and formalize the notions of context and collocations. Addressing these challenges requires expertise from both philosophy and computer science.

To overcome these challenges and help to design and implement the functionality of a distant reading system, we see the design of a benchmark as a next important step. A benchmark is a set of operations that forms the basis to measure and compare the performance of different software implementations [19]. For instance, benchmarks are prominently used in the field of databases to compare different implementations of SQL. They implicitly define the functionality, help to identify performance bottlenecks, and enable meaningful comparisons of system implementations. Other examples, recently published in the field of digital libraries, are benchmarks on author disambiguation [43] and plagiarism detection [44].

We deem our benchmark *user-oriented*, since we focus on the requirements of a specific user group, conceptual historians. This means that our benchmark represents the expected workload of conceptual historians working with a distant reading system, i.e., what types of queries they use. In the end, it enables two ways of evaluation: First, our benchmark assesses the feasibility of distant reading from a user perspective. It allows studying how results depend on characteristics of the corpus, such as its size. To illustrate, one might ask how the sizes of the collocation sets for a specific word, say "democracy", differ when computed on corpora of different sizes. One would run the same query on a large corpus, like the Google Ngram data set, and on a much smaller corpus, for instance one which conceptual historians have already studied exhaustively intellectually. This is important, because, in the end, results need to be interpreted by a human expert. Second, the benchmark allows measuring the performance in terms of run time and helps to find bottlenecks and improve specific implementations. This needs to be independent of the technology, e.g., whether a system is built upon a relational database management system (RDBMS) or a MapReduce framework. The design of the benchmark sketched so far is the topic of this current article.

More specifically, we make the following contributions. Firstly, we collect and structure the information needs of conceptual historians and formalize the notion of word context in conceptual history. To do so, we rely on the four principles of word context introduced by Heringer in [18], a seminal piece of work in the field of corpus linguistics [42]. Due to the relevance of his work, we focus on his four principles of word context: *time*, *search radius*, *frequency*, and *affinity*. Our contribution here is to formalize these principles. This allows representing the information needs of conceptual historians. Secondly, we have compiled a list of design decisions behind the benchmark, and we explain and justify our respective choices. Thirdly, we propose an actual benchmark. It contains queries that mimic typical ways of conceptual historians discovering scientific information. More specifically, we have come up with query templates reflecting the anticipated load a conceptual historian would create using the system envisioned. Finally, we run our benchmark and conduct an evaluation, with two objectives. The first objective is to obtain insights regarding the content of a corpus. We look at the differences between a large and broad library (world literature) and small and very specific library (expert literature), the collected works of the philosopher John Stuart Mill in our case. One important result is that there appear

to be different perspectives on Mill's research topics in different corpora. To take in these perspectives, we discover information across multiple repositories. The second objective is to benchmark two different technologies: an RDBMS and a MapReduce framework. We observe that row-based database technology often provides lower response times than modern MapReduce frameworks. We think that this is mainly due to more sophisticated indexing functionality with the first alternative.

Paper outline: Sections 2, 3, and 4 feature fundamentals and related work, from different perspectives. Section 6 covers information needs in the field of conceptual history. Sections 7 and 8 feature formalizations of the word context, i.e., we formalize the computation of collocation sets based on a text corpus and propose operations on these collocation sets to facilitate the interpretation of context. We explain the design decisions behind our benchmark in Section 5 and describe its queries in Section 9. Section 10 features our evaluation. Section 11 concludes.

2 Related Work

Applying computational techniques to traditional humanities problems is called *digital humanities* [5]. This includes using data analysis methods in various humanistic disciplines. In this section, we review approaches, solutions, and data sets used in digital humanities to analyze large text corpora. In the next section, we describe fundamentals of the subfield of conceptual history.

Distant Reading

Applying computational methods on literature data or digital libraries is known as *distant reading* [31]. Distant reading is a collective term referring to a range of computational methods, analyses, and library data. One example of distant reading is to provide insights regarding linguistic word usage at a statistical level. Hamilton et al. [17] propose the law of conformity that infrequent words are more likely to change their meaning than frequent ones. Another example is to analyze the importance of topics, e.g., of scientific ones [36].

Culturomics

Culturomics is the study of human language, culture, and behavior by analyzing digital texts.[1] A popular example is the analysis of the evolution of the English-speaking culture based on the text printed in books [28]. Another example is the analysis of user-related content and user interactions in social networks to study culture changes [27].

Language Models

Language models are probability distributions that statistically model properties of natural language, e.g., the likelihood of a sequence of words in the English language. When focusing on the semantics of words, there are word embedding models like Word2Vec [29], GloVe [35], and BERT [7]. Word embedding models aim to capture the contextual meaning of words. To this end, these models learn a projection from a word to its surrounding words (skip-gram) or the other way around (continuous bag-of-words). Both methods result in an embedding representation where each word is represented by a vector in a high-dimensional vector space.

There are several kinds of information needs where a word embedding model can be helpful. First, one may be interested in words that are used statistically similarly to a word in question, i.e., querying synonym words. A second information need is to quantify the similarity between two words in question [8]. To deal with both kinds of information needs, word embeddings use surrounding words to determine the position of each word in the vector space [29]. The more similar the surrounding words, the closer are the positions of the word projections in the vector space. However, conceptual historians are interested in the question why the meaning of a word has changed. Thus, the information need is to find indications for a change of meaning in text. This calls for a comprehensive analysis of the surrounding words for a word in question rather than querying words that the embedding model has learned to be similar.

In addition, word embedding models provide a way to analyze changes in the meaning of words over time. For this purpose, one trains two models: one with text written in the present time

[1] This definition is from the Cambridge Dictionary: https://dictionary.cambridge.org/us/dictionary/english/culturomics.

and one with text written k years ago. When comparing both models, one can query for words whose meaning, i.e., whose surrounding words, has changed [9]. However, one cannot query for the reasons of a change or analyze a particular change in more detail. More precisely, one cannot answer the questions which surrounding words have caused this change, or if these surrounding words have been added or removed from the context. Furthermore, word embedding models can only be queried for the points in time that have been chosen at training time.

Latent Semantic Analysis

Latent semantic analysis (LSA) [6] and its subsequent approaches [37] essentially use singular value decomposition (SVD) to perform principal component analysis on documents, i.e., on a word by document co-occurrence matrix. Each principal component identified is interpreted as a topic of the documents. This allows finding similar documents based on similar principal components, i.e., their main topics. However, information needs of conceptual historians tend to be *word-centered*, i.e., they are interested in the contexts of words and their changes over time. This is different from information needs against documents other users might have, i.e., finding documents containing certain information or documents being similar to a given "query document".

Analyzing the document level is one application of LSA, i.e., one applies SVD to a word-document matrix [37]. One can also apply this technique at the word level. This means applying SVD to a word-by-context matrix. The word-by-context matrix contains the frequency of each word in each text window of, say, 7 words [23]. The resulting vectors represent the principal context of the words [22]. Word embedding models created in this way are subject to the same limitations as the other word embedding models described previously.

Text Corpora

When analyzing data to study human behavior, the selection of the data is crucial. We already mentioned the Google Books Ngram Corpus, one of the world's largest collections that includes a large fraction of all books ever published. Next to it, there exist other very large temporal text corpora, like HathiTrust, the Internet Archive, or Twitter data sets. HathiTrust in particular has an active community that works with the corpus and continuously extends it. For example, there is an additional data set that provides metadata and preprocessed feature extraction for the corpus [33]. However, we had decided to focus on the Google Books Ngram Corpus, since it is most popular and well known in the humanities and digital humanities community.

Query Workload on Corpora

There exist systems or query languages [26; 1; 39; 40; 34; 46] to deal with temporal data and even text corpora annotated with temporal information. But it currently is unclear how useful they are for conceptual history, as well as how to assess this. In addition, it is unclear how to simulate a typical workload for studies in the field of conceptual history.

3 Fundamentals

In the following, we provide some background regarding conceptual history. We do this for two reasons. Firstly, we want to ease understanding of the use case itself. This includes a fundamental issue that conceptual historians try to solve with distant reading [31]: small sample sizes in current, "manual" research processes. Regarding this issue, digitization might provide a new perspective. Secondly, we outline how conceptual historians tend to work, and which kinds of information are of interest here. This serves as a motivation for various features of the system envisioned.

3.1 Conceptual History

Conceptual historians study how the meaning of concepts, represented as words, evolves over time. Uncovering and understanding such changes then allow to model language changes, which in turn tend to be interpreted in how far they reflect societal developments [16; 24; 21; 15]. Conceptual historians focus on words with a high degree of abstraction, like "war", "peace", or "democracy".

Example 1 Think of the word "democracy". Democracy refers to a political concept implying, among others, that the population elects political leaders. Comparing today's interpretation with the one in

Ancient Greece, we observe that population (i.e., who may vote) is interpreted differently. For instance, in Ancient Greece, it did not refer to women. Based on such changes, a conceptual historian draws conclusions regarding changes in society, reflecting the cultural evolution of mankind.

3.2 Digital Conceptual History

Conceptual history is a good candidate for digital analysis because studies in this field primarily deal with texts and words [20]. John Rupert Firth [10] has observed that: "You shall know a word by the company it keeps." This has given way to the following axiom.

Axiom 1 *The essential meanings of a concept are reflected by how it is used in the context of other words.*

This axiom implies that one can extract collocations which reflect the historical semantics of a concept from written texts. In other words, one can derive the historical semantics of a concept, e.g., democracy in Ancient Greece, only by studying texts from the periods in question [15]. This is known as Koselleck's assumption to develop the field of conceptual history [20].

Syntagmatic Relations

Examining a word's historical semantics requires considering text from different points in time. Linguists describe evolutionary parts of language as diachronic [38]. To capture the semantics of a word, one has to consider text units like sentences, text fragments, or ngrams the word is used in [10; 18; 14; 13; 17].

Definition 1 (Syntagmatic Relation) The syntactical positioning of two words in texts creates a relationship between them, the *syntagmatic relation* [4].

A syntagmatic relation implies that the relationship between a word and other words is based on the syntax of the underlying written texts. This means that, when studying syntagmatic relations, experts only rely on written texts. Thus, one can extract syntagmatic relations from any kind of written text, e.g., from digital libraries.

Example 2 This example focuses on the syntagmatic relations of "coffee". Think of the text fragments "a cup of hot coffee" and "Coffee or tea?" One syntagmatic relation is that "hot" is used before "coffee", i.e., the adjective is used before the noun. The syntax of the English language defines this. Another syntagmatic relation is between "coffee" and "tea".

Collocations

Barnbrook at al. [3] have observed that there is more than grammatical and syntactical information in language. There also exist relations between words that co-occur in speech and text. Such a relation is a collocation. See Example 3.

Example 3 Barnbrook et al. [3] analyze the relation between "strong", "powerful", and "argument". Adjectives "strong" and "powerful" are in the same grammatical class. But an English speaker prefers "strong argument" over "powerful argument". Collocations capture such non-syntactical information.

Collocations are a key element to analyze the word context. We use collocations frequently in the following and will give a formal definition later. At this point, we limit ourselves to a brief description: To obtain collocations, conceptual historians specify a key word in context and collect the words immediately surrounding it. The resulting set of words gives conceptual historians an idea of how words are used. Building such a set of collocation from a corpus is called collocation extraction.

3.3 Small Sample Sizes

We now outline an issue, controversially discussed in conceptual history for half a century, which can be addressed only by digital analysis. Today, research in conceptual history means to *manually* read literature from the time under investigation. The method is that a human reader locates relevant concepts and studies the respective syntagmatic relations, i.e., close reading [31]. This means that knowledge on conceptual history is often based on few publications that are deemed standard literature [21]. These are, say, articles written by researchers of that time. Even if the literature is well chosen, it is questionable whether one can draw general conclusions from a small sample of books. This may lead to a filter bubble, well known

from today's social networks [41; 11; 2]. To arrive at new insights and to prevent a filter bubble issue, one must examine a large part of the world's literature. Due to limited human reading speed, this is only possible with support by technical systems.

4 A Query Algebra for Conceptual History

The benchmark proposed in this article is not tailored to a specific query language or system implementation. However, to define it, an adequate representation of the queries is needed. For this purpose, we now briefly review CHQL [46], a query algebra that has been designed to formulate information needs from conceptual history. It targets what we call temporal text databases. Its specification not only consists of definitions of algebraic operators, but also of the underlying structure, i.e., a data model. Regarding the data model, the core notion is a tuple, but its definition is different from the conventional, relational one. Each tuple represents a different *ngram*, i.e., a sequence of n words. In addition, each tuple includes an array containing the usage frequency of its ngram over time. Formally, a tuple is `Ngram(ngram: string, counts: long[])`. Based on this data model, CHQL features operators to formulate information needs. CHQL contains (1) simple operators, e.g., to select elements based on the ngram text, (2) temporal operators, e.g., to search for elements with a similar usage frequency, which are represented as time series, and (3) linguistic operators, e.g., to search for words that appear together (cooccur). One example of a linguistic operator is *surroundingwords*. It compiles a set of all words that are used around a target word. One can see this as an initial approach to catch the context of a word. In general, the CHQL algebra allows expressing queries like:

- What are the nouns with a usage frequency larger than 10,000 in year 1950?
- What is the number of surrounding words for "east" in the 20th century?

We see CHQL as a means to implement distant reading. See [46] for a complete description. In this article, we focus more on distant reading and on analyses of word context than in [46]. We will provide a comprehensive view on word context, develop a respective formal definition and use it to build our benchmark.

5 Design Decisions Behind our Benchmark

So far, we presented some basics on distant reading and conceptual history. Before going into the details, we justify the objectives and fundamental design decisions behind our benchmark. The objectives of our benchmark are as follows.

Corpus Comparison. One objective is to provide insights into the content of a corpus and to facilitate statements related to its content. This is the application-specific benefit of our benchmark, i.e., the added value to conceptual historians.
Performance. Another objective is to specify queries to measure and compare the run times of implementations of distant reading systems. This is the technical benefit of our benchmark.

Following these objectives, we make some design decisions regarding our benchmark. We see these decisions and their writeup as another contribution of this article. We present and discuss them in the remainder of this section.

5.1 Query Templates

Our first design decision is whether our benchmark consists of queries or of query templates.

- Hard-coded queries are static and ensure maximum comparability of the systems investigated.
- Query templates are templates of a query that a one can execute many times with different parameterization, to benchmark certain operators in a specific order.

For our benchmark, we have opted for query templates, for two reasons. First, query templates allow one to execute any number of queries, for comprehensive tests of the system. Second, they facilitate customization of the benchmark by specifying the parameter space, e.g., analyze words from a specific subject area or from a certain dictionary from conceptual history.

5.2 Mapping Information Needs

Our second design decision is how to simulate the information needs. The alternatives are the following:

- One query template simulates various information needs, since the information needs build on each other.
- One query template simulates exactly one information need, to evaluate the performance of queries for different information needs.
- The benchmark defines a number of query templates, to simulate a single information need to evaluate query performance in a broad manner.

For our benchmark, we define a single query template for each information need. The queries to satisfy one information need are fairly similar for distant reading. This means that when we identify, say, four information needs, our benchmark will consist of four query templates.

5.3 Query Results

The third design decision has to do with the structure of results. We see the following alternatives:

- Leave the structure of the result of a query template open, i.e., results of any structure are allowed, in order to evaluate as many operator combinations as possible.
- Each query template returns a set of collocations.
- Each query template includes an aggregation operation, to yield results with a specific size.

We decide to let each query return a collocation set. This is for two reasons. First, collocation sets are in the center of interest of distant reading systems. Results other than collocation sets are incidental, since they do not yield any additional information in our use case. Second, a uniform structure of all results allows for better comparability of the results. For instance, it may be interesting to compare the size of results of different query templates.

5.4 Data Set

Our next design decision has to do with the data.

- Specify the data set to ensure maximum comparability of the test systems, i.e., specify a particular corpus.
- Specify the schema of the data set to allow evaluating data sets of several sizes and with several data characteristics, i.e., allow any temporal text corpus.

We decide to specify the schema of the data, but not a particular corpus. Regarding the first objective listed earlier, our benchmark allows to compare the query results on several corpora and to make statements about the content of a corpus.

5.5 Algebraic Formulation

Our last design decision is the query language or formal language to specify the query templates. We see the following alternatives:

- Formulate the query templates in a widely used query language, like the Structured Query Language (SQL). This will result in lengthy, complex query statements.
- Formulate the query templates in a special query language, like CHQL [46].
- Provide a mathematical formulation of the query template in form of an algebraic expression.

We decide to provide the query templates of our benchmark as algebraic expression since there is no widely used query language for distant reading systems.

6 Information Needs

Before we define our benchmark queries from a technical perspective, we motivate why these queries are relevant from the user perspective. A benchmark with a random assortment of queries does not allow to draw conclusions from its result. To that end, we first identify relevant information needs and then derive queries from them. In this section, we describe information needs coming from conceptual history.

6.1 Identifying Information Needs

To identify information needs in conceptual history, we, on the one hand, have surveyed relevant literature (see Section 3) and, on the other hand, rely on expert knowledge. We have become familiar with these information needs by interacting with practical philosophers who are part of our organization (KIT), and with whom we have been

collaborating for several years. We performed our survey according to the well-established systematic by Webster and Watson [45]. Roughly speaking, this method systemizes forward and backward steps in literature search to illuminate a subject broadly and regarding the current state-of-the-art.

6.2 From Text to Meanings of Words

We now describe the information needs of conceptual historians to derive the meanings of words. Roughly speaking, conceptual historians are interested in the following information:

- Selecting syntagmatic relations of a target word.
- Build a set of collocations of a target word.
- Filter the set of collocations regarding the object of investigation, e.g., filter for nouns or for philosophical words.
- Compare collocation sets with each other.

We found that the information needs of conceptual historians build on each other. So we structure the information needs in levels. For instance, the first level contains syntagmatic relations of words in text. Each level uses information from the previous level.

Table 1 shows the information needs where each row corresponds to a level. In each row, the table has the following entries:

Level. A unique number to identify the level.
Name. Our name for the information need.
Linguistical description. A concise description of the information need from the perspective of a conceptual historian.
Technical transformation. A description of the necessary transformation from the previous level to the current one.
Information structure. The format of the data to cover the information need.
Example. An example of the result for the information need. [2]

We see the table and the structure of the information needs as one contribution of this article.

In the following, we first describe the particularities of the first and the last level. We then describe the analogy between Table 1 and a human reader when doing close reading. Sections 7 and 8 cover the specifics of the transformations.

First and Last Level

Level 0 stands for the corpus, i.e., the data set. Level 5 is the interpretation by conceptual historians. Both Level 0 and Level 5 actually are not information needs, but we need them to cover our use case. Level 5 indicates that a distant reading system supports conceptual historians and does not target at replacing them. When knowing the meaning of a word, it is the intellectual effort of a conceptual historian to identify changes in meaning and how they reflect cultural changes.

6.3 Analogies with a Human Reader

A human reader selects a set of books or texts to determine the meaning of a word in question (target word). To do so, she focuses on paragraphs and sentences that use the target word (Level 1). When reading the selected text snippets, a human reader can infer the meaning of the word from the context in which it is used (Level 2). I.e., one implicitly analyzes the context of a word by identifying how the target word interacts with other words close to the target word. Here, a human reader neglects irrelevant words like stop words and only captures informative words like nouns or verbs (Level 3). The distinction between irrelevant and informative words depends on the reader as well as on the object of investigation.

When the historical or current meaning of a concept is known, the task of a conceptual historian is to determine whether the meaning has changed in a certain period. To this end, she determines the meaning at different points in time (Level 4). See Example 1. According to Axiom 1, such a change is visible when studying the context of democracy in the given time period [4; 15].

Consequently, even without fully understanding all aspects of the meaning, it is possible to indicate changes of meaning, by analyzing whether collocations are added or omitted—either as a human reader or with a distant reading system.

6.4 Analogies with Data Mining

Table 1 describes the data transformations that are necessary to meet the specific information needs. To complete the presentation, we map the

[2] The examples are inspired by Alexander Friedrich and Chris Biemann. *Digitale Begriffsgeschichte?: Methodologische Überlegungen und exemplarische Versuche am Beispiel moderner Netzsemantik* [13].

Level	Name	Linguistical description	Technical transformation	Information structure
0	Input data	Digitized book inventory	Text corpus with temporal information	Set of ngrams, their usage frequencies, and part of speech annotations
Example:	*Google Books Ngram Corpus*			
1	Select syntagmatic relations	Focus on the word of investigation (target word)	Select ngrams that contain the target word (cf. Section 7.1)	Set of ngrams, their usage frequencies, and part of speech annotations
Example:	*Syntagmatic relations of "network":* { *"cellular networks", "network of railways", "network of relationships", "power network", "rail network", "road network", "sales network", "social networking", "water supply network"*}			
2	Determine contexts	Find words that relate to the target word, i.e., collocations	Split ngrams into single words (cf. Section 7.1)	Set of words and their frequencies of occurring with the target word
Example:	*Words around "network":* { *"cellular", "of", "power", "rail", "railways", "relationships", "road", "sales", "social", "supply", "water"* }			
3	Customize collocations	Specify collocation details depending on the object of investigation	Filter, count, aggregate, and group the set of surrounding words (cf. Section 8.1)	Set of words and their frequencies of occurring with the target word
Example:	*Remove stop words and filter on nouns:* { *"power", "rail", "railways", "relationships", "road", "sales", "supply", "water"* }			
4	Compare collocations	Identify collocation changes by comparing collocations for different target words or for different points in time	Set operations on two sets of surrounding words (cf. Section 8.2)	Set of words and their frequencies of occurring with the target word
Example:	*Intersection of the collocations for the words "network" and "infrastructure":* { *"power", "rail", "railways", "road", "supply", "water"* }			
5	Experts interpretation	Conclude or reason a observation	Result presentation	Plot, list, or graph of surrounding words
Example:	*A graph of surrounding words in which the frequencies specify the distances between the nodes.*			

Table 1 Information needs of conceptual historians structured in several levels.

levels to processing steps from the well-known data mining processing chain. Levels 1 and 2 perform *feature selection*, i.e., selecting the relevant features for a certain task. Levels 3 and 4 implement *data analysis*, i.e., carrying out the actual operations on the previously selected features. We organize the following chapters analogously: Section 7 describes relevant features. Section 8 is about the actual analysis.

7 A Formalization of Context

To capture the meaning of a word, we now formalize the notion of context. This formalization is another contribution of us. It is a realization of the information needs of Levels 1 and 2 of Table 1.

7.1 A Formal Definition of Context

We split the formalization into two steps, corresponding to the two levels in Table 1.

Level 1 is to locate relevant syntagmatic relations. *Level 2* is to extract the context of a word in the form of collocations.

In a digital corpus, one can access arbitrary text fragments. However, only text fragments which include the word under investigation contain syntagmatic relations for this word. In Step 1, we select all text fragments that contain the target word. In Step 2, we split each text fragment into individual words and select all words that occur closely to the examined word. This results in the *collocations* of the word in question.

Definition 2 (Context) The context of a word is the set of words surrounding it.

A word may have more than one context, depending on the text source and the specific mappings, i.e., the objects of investigation. We now give a formal definition of collocation sets.

Corpus and Reference Corpus

Natural language consists of utterances, as follows.

Definition 3 (Utterance) An utterance is a unit of speech, like a sentence or a text snippet.

$$utterance = \ldots\ word_{i-2}\ word_{i-1}\ word_i\ word_{i+1}\ word_{i+2}\ \ldots \tag{1}$$

Our starting point to formally define collocations is a hypothetical set A^* that contains all utterances of humans. This includes all past, present, and future utterances—independent of whether they are written, spoken, or thought. Even if one cannot explicitly compute this set, the idea is that it conceptually exists. A, a subset of A^*, is the set of utterances accessible to us, e.g., written text, sound recordings, etc.

Next, D is the set of correct utterances. This is a subset of the previous two sets. Here, *correct* means the correct use of language, allowing to discard, say, typos.

Definition 4 (Corpus) A corpus $C \subset D$ is a collection of books or other media.

$$C \subset D \subset A \subset A^* \tag{2}$$

In our case, C, as a true subset of D, corresponds to, say, the Google Books Ngram Corpus. The set of all references that can be extracted from C is the reference corpus (RC) [15].

Definition 5 (Reference Corpus) A reference corpus for word *word* is a corpus that contains only *utterance* that contain the particular *word*.

$$RC_{word} = \{utterance \in C \mid word \in utterance\} \tag{3}$$

Since all utterances follow the language syntax, a corpus RC_{word} contains all syntagmatic relations of the word *word*.

Example 4 Take the reference corpus $RC_{emancipation}$ for "emancipation". Syntagmatic relations are:

- The emancipation of the women ...
- ... order the emancipation of slaves.
- ... freedom as result of emancipation ...

Collocations

In order to get a better perspective of the context of a word, it is worthwhile to look at an aggregated overview of these references, as follows:

Definition 6 (Collocations) The surrounding words of *word* in RC_{word} are split into single words, i.e., 1-grams. This forms its collocations $RCOL_{word}$.

$$RCOL_{word} = \{word_i \mid word_i \in RC_{word}\} \setminus \{word\} \tag{4}$$

For example, the collocations of "emancipation" from the reference corpus $RC_{emancipation}$ are $RCOL_{emancipation}$. For the above example, the set $RCOL_{emancipation}$ contains the words "women", "slave", and "freedom", next to others.

Locating syntagmatic relations and mapping them to collocations are application specific, i.e., depends on the object of investigation. For example, a conceptual historian might not be interested in all collocations of a word, but only in the ones in a certain time period, say, the 20th century. This illustrates the need for temporal information of a collocation sets. We propose a more sophisticated definition of *context* in the next section.

7.2 The Dimensions of Context

To mimic Heringer's four principles, we now describe four dimensions to quantify the relationship between a target word and its surrounding words: time, search radius, frequency, and affinity. These dimensions control which surrounding words are deemed collocations and, thus, are relevant for the meaning regarding a specific investigation. We call them the dimensions of context.

Time Dimension

Conceptual historians are interested in changes of syntagmatic relations over time, i.e., to limit the corpus C to utterances used at the time of interest. This is basic functionality, allowing to detect the appearance or disappearance of meanings over time in the form of collocations. One can then relate what has been written to historical and cultural trends [28]. To this end, we extend our definition of context with the temporal dimension.

Definition 7 (Temporal Corpus) C^t confines the corpus to a given time interval.

$$C^t \subseteq C \tag{5}$$

For example, $C^{1920-1945}$ is a corpus containing sources from 1920 to 1945.

Based on this corpus, we define a temporal reference corpus and a temporal collocation set.

Definition 8 (Temporal Reference Corpus)

$$RC^t_{word} = \{utterance \in C^t \mid word \in utterance\} \tag{6}$$

$$RC^t_{word} \subseteq RC_{word} \tag{7}$$

Definition 9 (Temporal Collocation Set)

$$RCOL^t_{word} = \{word_i \mid word_i \in RC^t_{word}\} \setminus \{word\} \tag{8}$$

$$RCOL^t_{word} \subseteq RCOL_{word} \tag{9}$$

These are sets of syntagmatic relations that have occurred over a period of time.

Search Radius Dimension

Apart from the temporal dimension, the context of a word consists of words used closely to it. This current dimension defines *close*. Formally speaking, the search radius r specifies the size of the window whose words are part of the collocation. So, in addition to the temporal dimension, the reference corpora and collocations depend on r.

Definition 10 (Spatio-Temporal Reference Corpus)

$$RC^{t,r}_{word} = \{(word_{i-r}, \ldots, word_i, \ldots, word_{i+r}) \mid \\ \mid (\ldots, word_{i-r}, \ldots, word_i, \ldots, word_{i+r}, \ldots) \in \\ \in RC^t_{word}\} \tag{10}$$

Definition 11 (Spatio-Temporal Collocation Set)

$$RCOL^{t,r}_{word} = \{word_i \mid word_i \in RC^{t,r}_{word}\} \setminus \{word\} \tag{11}$$

Heringer defined the radius to be the same for the forward and backward window. In principle, they can have different sizes for collocations before and behind the target word. For the rest of this article, the radius is according to Heringer, i.e., same radius r for both windows.

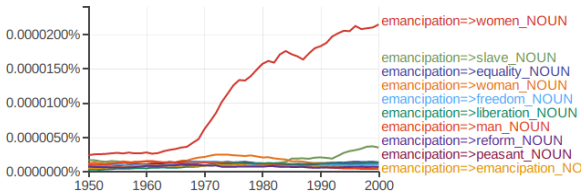

Fig. 1 Collocations of the word "emancipation" over time and filtered on nouns.[3]

Frequency Dimension

To gain an indication how *important* a specific collocation is, we propose a weighting factor for each collocation. The weight depends on two dimensions: usage frequency and affinity.

The intuition behind the frequency dimension is that the most frequent collocations at time t reflect the primary meaning of the word in question at t. Frequency in combination with time forms the foundation for diachronic studies by conceptual historians [4; 14; 18].

To include the frequency, we first extend our definition of a corpus. We add the frequency of an utterance to the data model of the reference corpus $RC^{t,r}_{word}$ as well as of the collocations $RCOL^{t,r}_{word}$. This results in three-tuples of $(utterance, t, freq)$ and $(word, t, freq)$, respectively. The frequency of syntagmatic relations gives way to weighting collocations.

Example 5 A conceptual historian studies how women's movements have influenced the meaning of the word emancipation. Her hypothesis is that a relationship with "women" dominates the meaning of the word "emancipation". She obtains Figure 1. This strengthens her hypotheses. Note that the example is over-simplified since the expert only consults the time dimension with a fixed weight on the frequency. In a more realistic example, she would also consider other dimensions like the affinity of both words as well.

Affinity Dimension

Affinity describes the proximity of a collocation to its target word. Besides frequency, this is the second weight dimension that indicates the *importance* of a specific collocation. For example, in the syntagmatic relation "the emancipation of women", "women" has an affinity of 2 to the target word "emancipation", since it is syntactically used within 2 words. The affinity is the same whether the collocation is used before the target word or after it. Words close to each other are expected to share a higher affinity than distant ones [18; 12; 13].

Target words and their collocations are not always used in the same syntactical proximity. In some utterances, a surrounding word occurs with a distance of, say, 2, in others with another distance. To get an overall distance, we define affinity as the average distance over all utterances.

Example 6 A conceptual historian studies the collocations of "emancipation" in 1974. An affinity value of 2.7 means that "women" occurred with an average distance of 2.7 words around "emancipation".

Summary

The four dimensions time, radius, frequency, and affinity are different ways to specify the mapping from syntagmatic relations to collocations. This allows one to create the context of a word and also different user-specific views. Such views might be the context of a word in a certain period of time.

7.3 Example Queries

We now illustrate information needs of conceptual historians. We use information needs like these to define the queries in our benchmark.

Example 7 A conceptual historian wants to have a look at the collocations of the word "emancipation".

Example 8 A conceptual historian is interested in the collocations of the word "censorship" in the first half of the 20th century.

Example 9 To study changes in the usage of geographic directions [46], a conceptual historian requests the collocations of the words "east" and "west" with a radius of 4 words.

[3] The Google Ngram Viewer shows this syntagmatic relations at https://books.google.com/ngrams/graph?content=emancipation%3D%3E*_NOUN&year_start=1800&year_end=2000&corpus=15&smoothing=3.

8 Preparing Collocation Sets for Interpretation

In the previous section, we formalized the context of a word by finding syntagmatic relations (Step 1) and identifying collocations (Step 2). According to preliminary experiments of ours, Step 2 often results in collocation sets with its hundreds or thousands of words. This is too much for users to analyze manually. To support experts to determine the meanings of a word, we split the analysis of collocation sets into the following two steps.

Step 3 is to filter and aggregate collocation sets.
Step 4 determines differences to reference points, by comparing a collocation set with other ones.

Both steps reduce the volume of data, by focusing on information relevant to the user. In this section, we motivate how to reduce the data and then say how to perform Steps 3 and 4 using a system.

8.1 Filter and Aggregate Collocation Sets

Perceiving the usage frequencies per year as a 2D matrix, i.e., a row contains the frequencies of a certain word, we see two ways of reduction. Firstly, there is *filtering* to remove rows or columns. Secondly, there is *aggregating* to combine multiple rows or columns to a single one. We explain both operations in the following.

Filter Functionality

Filtering collocations only keeps relevant collocations regarding the object of investigation. Several kinds of filter are required.

Text filter. Filter words and text fragments using regular expressions.
Weight filter. Filter collocations based on their weights, e.g., their usage frequency.
Part-of-speech filter. Filter corpus-included word annotations, e.g., on the part-of-speech of a word.

Aggregate Functionality

One can apply aggregation either horizontally or vertically.

Horizontal application means to combine the usage frequency over a period, e.g., the usage frequency within a decade or century.

Vertical application means to combine the frequency values or weights for all collocation of a single year, e.g., the year 1899.

According to our formalization of context in Section 7, the following aggregate functions are relevant: sum, average (i.e., arithmetic mean), min, and max. The semantics of these functions are the usual ones, cf. [1].

8.2 Comparing Collocation Sets

There are three types of comparison that are of interest to conceptual historians.

Intersection creates the common context of two words.
Union creates a context over several words, e.g., "north", "east", "south", and "west".
Minus removes specific collocations from the context, e.g., for ambiguous words.

Example 10 A conceptual historian studies changes in the meaning of the word "emancipation" between 1950 and 2000. In other words, she is interested in collocations that occur in that time. To find them, one generates one collocation set for "emancipation" at 1950 and one for 2000, i.e., $RCOL^{1950}_{emancipation}$ and $RCOL^{2000}_{emancipation}$. To find the desired collocations, one can subtract the collocations of the year 1950 from the ones of 2000, i.e., $RCOL^{1950}_{emancipation} \setminus RCOL^{2000}_{emancipation}$.

8.3 Example Queries

To illustrate further, we now show some example information needs. Examples 11 and 12 correspond to Level 3 of Table 1. Examples 13 and 14 are information needs on Level 4.

Example 11 One information need is to find the collocations most frequently used with "emancipation" in the period from, say, 1930 to 1990. This includes the sum over this period as well as the average.

Example 12 A conceptual historian is interested in the *topics* the word "east" is used in, rather than the collocations themselves. Experts expect to see topics like geography, politics, and military and are interested in how pronounced they are.

Example 13 One is interested in the common context of words "emancipation" and "women".

Example 14 One is interested in the context of "mouse" at the end of the 20th century which it did not have a hundred years earlier.

9 Our Set of Benchmark Queries

We now present the actual query templates that make up the benchmark. The query templates, which we describe subsequently, are: (1) collocation selection, (2) horizontal aggregation, (3) collocation grouping, and (4) set comparison.

For better readability, we first explain the role of the parameters of each template, then give examples and describe concrete instantiations. Query instantiation is the step from the query template to an actual executable query, i.e., the parametrization of the template. One can instantiate each template arbitrarily many times and customize these queries in various ways, by specifying their parameters by hand.

9.1 Query Template: Collocation Selection

The collocation selection query template queries the surrounding words $RCOL^{t,r}_{word}$ of some word *word* at time t within a radius of r (cf. Equation 11). This template benchmarks the system's property to filter relevant parts of the context and to project them to collocations. The query template has the following form:

Query Template 1 Collocation Selection

collocation (
  word : string,
  $(from, to)$: time interval,
  r : integer,
  filter : filter predicate
) := $RCOL^{(from,to),r}_{word}$

We describe the parameters in the following:

Word. This parameter is a literal word, a list of words, or a regular expression. In case several words are given, the result is the union of the individual collocation sets.
(from, to). This tuple specifies the desired time interval. All utterances whose time labels t satisfy $from \leq t \leq to$ are selected.
Radius r. This parameter specifies the number of words before and behind the search word.
Filter predicate. This optional parameter allows applying filter functions of Section 8.1.

Example Query

The following query instantiates Example 7.

```
collocation ("emancipation", (1800,2000), 5);
```

Query Instantiation

When creating queries from this template, we randomly select words from the corpus with uniform probability. Next, we draw two random time labels where the smaller one becomes the value of *from* and the larger one the value of *to*. Finally, the radius is drawn uniformly between 1 and the largest radius possible, i.e., the largest ngram chain in the corpus.

9.2 Query Template: Horizontal Aggregation

This template generates queries to benchmark the capability to do horizontal aggregation. The aggregate can depend on the frequency of the collocation, on the proximity of a collocation (affinity), or on both (cf. Section 7.2). The template has the following form:

Query Template 2 Horizontal Aggregation

conflate (
  col : collocation,
  map : map function,
  reduce : aggregate function,
  order : sort predicate
) := *col* (specific weights and sorting)

We describe the parameters in the following:

Collocation. We use the first template to generate collocations.
Map function. This parameter specifies how to compute the value used in the aggregate step,

i.e., it maps each collocation to a value. One can either directly use the frequency (`FREQ`) or affinity (`AFFI`) values or freely define a function which may consider both values.
Aggregate function reduce. This parameter specifies the aggregate function to use, e.g., `SUM` or `AVERAGE`.
Order. This is an optional parameter that specifies whether to order the result according to some criterion. The default is to sort by the weight value in descending order, while `null` disables sorting.

Example Query

The following query represents the question in Example 11.

```
conflate (
    collocation ("emancipation", (1930,1990), 3),
    FREQ,       // use frequency per year
    SUM,        // aggregation function
    DESC);      // sort in descending order
```

Query Instantiation

The parameters of the collocation template are selected as explained before. For the map function, one of the three following function is drawn with equal probability: (1) `FREQ`, (2) `AFFI`, or (3) `FREQ` · `AFFI`. The reduce function is selected randomly among: `SUM`, `COUNT`, `MIN`, `MAX`, `AVERAGE`. Finally, with a probability of 0.5, the result is sorted according to the weight value. With a probability of 0.1, the query specifies to sort the collocations in a lexicographical order. Otherwise, with a probability of 0.4, no sorting takes place.

9.3 Query Template: Collocation Grouping

The next template aims at benchmarking the grouping of collocations and subsequent vertical aggregation of the temporal weights. This represents Example 12, i.e., a conceptual historian who studies groups of collocations aggregated as topics.

The required group keys usually are not part of the corpus. So we have to rely on an external source, i.e., a list of key-value pairs that provide group keys. Using an external source has the advantage to perform different content-related types of grouping, like topic grouping and sentiment grouping.

Topic Grouping. A topic list specifies a more general term as group key, i.e., the topic a word belongs to. For example, the words soldier, army, and tank belong to the topic military. We use a categorization list generated from OpenThesaurus [32] that contains 33 topics.
Sentiment Grouping. Using a sentiment list works similarly, except that it only has three groups: positive sentiment, negative sentiment, and neutral or no sentiment. We use the LIWC sentiment list [47] to join the sentiment group keys.

The template is as follows:

Query Template 3 Collocation Grouping

```
grouping (
    col     : collocation ,
    keys    : group key list ,
    reduce  : aggregate function
)           := col (grouped and aggregated)
```

We describe the parameters in the following:

Collocation. We use the first template to generate collocations.
List keys. A source list for the group keys.
Aggregate function reduce. This specifies the function to vertically aggregate the values of the time series, i.e., per year.

Example Query

The following query implements Example 12.

```
grouping (
    collocation ("emancipation", (1930, 1990), 5),
    "topic_mapping",   // group key list
    SUM);              // aggregation function
```

Query Instantiation

To instantiate queries from this template, we select the *keys* parameter randomly with uniform probability. If type sentiment is chosen, we use the LIWC sentiment list [47] as group keys. If type topic is chosen, we use the categorization list generated from OpenThesaurus [32]. As vertical aggregate function, `SUM` or `AVERAGE` is selected randomly.

9.4 Query Template: Set Comparison

The final group of templates are set operations on collocation sets, i.e., intersection, union, and set minus. The template has the following form:

Query Template 4 Collocation Set Comparison

compare (
*col*1 : collocation lhs,
setop : set operation,
*col*2 : collocation rhs
) := *setop*(*col*1, *col*2)

We describe the parameters in the following:

Collocation. We use the first template to generate collocations.
Set Operation. Specifies one of the three set operations: intersect, union, and minus.

Example Query

The following query formalizes the information need in Example 14.

```
compare (
    collocation ("mouse", (1971,2000), 4),
        minus,
    collocation ("mouse", (1901,1930), 4));
```

Query Instantiation

In addition to the instantiation of two collocation selection queries, the set operation is drawn from among the three set operations, with equal probability.

This template is also used to compare the meaning of a word in different corpora, to provide insights regarding their content.

9.5 Relationship with Table 1

Our query templates cover all information needs from Table 1. Query Template (1) collocation selection covers Levels 1 and 2 since it selects ngrams from the corpus and extracts a set of collocations. Query Templates (2) horizontal aggregation and (3) collocation grouping (i.e., vertical aggregation) cover Level 3. Query Template (4) set comparison covers Level 4.

10 Benchmarking Distant Reading Systems

In this section, we benchmark distant reading systems using our benchmark. This section has three parts. The first one describes the objectives of our evaluation. The second part describes our experimental setup. The third part assesses the informative value of query results for different corpora. In the last part, we test the run-time performance and try to identify performance bottlenecks.

10.1 Objectives

As mentioned, our evaluation has two objectives: (1) the informative value of query results and (2) performance benchmarking.

10.1.1 Informative Value

In our experiments, we study the following questions.

Objective: Result Sizes

To what extent does the number of collocations depend on the corpus size? In other words, how does the result size of our benchmark queries change with larger corpora?

Objective: Comparison of Corpora

John Stuart Mill is a well-known philosopher and one of the most influential thinkers of the 19th century [25]. To assess his influence on our society, we compare his works with the world's literature: To what extent do Mill's research topics differ in expert literature and world literature? Here, *world literature* is a collection of literary works with a wide popularity across national and regional boundaries that are deemed significant for the world population. In other words, whether something is world literature primarily hinges on its popularity. In contrast, *expert literature* are literary works that target specifically at a professional audience. For example, this is literature that consolidates the research of Mill or is about a specific scientific topic. The transition between world literature and expert literature is smooth. For example, there are works from the philosopher Mill that have become popular and, thus, are both expert literature and world literature.

Objective: Insights Regarding Content

How do results differ in terms of content between different corpora? When comparing the collocation set of the same word on different corpora, we seek insights regarding different perspectives on a word. For example, think of the collocation set of the word "mouse". Collocations in technical literature on computers might be very different from collocations in books on animals. Examining such differences also helps to quantify differences in perspectives in expert literature and world literature.

10.1.2 Run-Time Performance

To benchmark the run-time performance, we differentiate between selection and analytical queries. Template (1) collocation selection contains selection queries. Templates (2) horizontal aggregation and (3) collocation grouping in turn contain analytical queries. Observe that analytical queries have to select the data to be analyzed in the first place as well. Queries of Template (4) set comparison combine selection and analytical querying functionality. They do so by first selecting and extracting collocations and then comparing two sets of collocations.

This evaluation has two objectives: to give a first indication regarding the usefulness of existing technology for distant reading and to assess the soundness and helpfulness of our benchmark.

Objective: Comparison of Technology for Data Management

One may be interested in the performance of different technologies.

Objective: Verification of our Benchmark

Another objective is to evaluate our benchmark. We examine whether our benchmark, as well as its grouping of the queries, yield conclusive and helpful information on the efficiency of the two concrete systems tested. We expect RDBMS to perform better for selection queries and MapReduce to be faster on analytical queries. Since our benchmark simulates a typical workload, we can analyze which aspect is more important in a distant reading scenario. At the current level of analysis, it will already be interesting whether there are big differences regarding the run times for the different query templates.

10.2 Experimental Setup

We now describe the data sets used and the experiment setup.

10.2.1 Data Sets

In our experiments, we use two corpora, a small one and a large one. We explain our selection in the following and describe our preprocessing.

Motivation

So far, conceptual historians tend to use a comparatively small set of selected books for their investigations. To mimic this, we use the *Collected Works of John Stuart Mill* (JSM) as our first corpus. Mill is "the most influential English-speaking philosopher of the nineteenth century" [25] and well-known to conceptual historians. This corpus represents a typical amount of books a conceptual historian may read for an examination in conceptual history. The Collected Works of John Stuart Mill is expert literature.

As second corpus, we use the Google Books Ngram Corpus[4] (GBNC), a corpus from one of the largest book collections in the world. It contains more than 8 million books that, as a whole, have never been used for investigations in conceptual history. The Google Books Ngram Corpus contains world literature.

Data Sets

We now describe the data sets in more detail.

JSM. We build an ngram corpus from the Collected Works of John Stuart Mill. JSM is a small corpus that contains over 28,000 1-grams and 1.7 million 5-grams.
GBNC-full. We use the Google Books Ngram Corpus of the English language. It is one of the largest corpora openly available and is of interest to conceptual historians. It contains over 5 million 1-grams and nearly 318 million 5-grams.
GBNC-1mio. We created a sample of the full Google Books Ngram Corpus with random sampling. We do this for two reasons. First, having two corpora of different size, but with the same base, we can study which differences are due to corpus size. Second, we want to facilitate comparisons

[4]The Google Books Ngram Corpus is available at https://storage.googleapis.com/books/ngrams/books/datasetsv2.html.

between expert literature and world literature which are not blurred by different sizes of the corpora.—Just downsampling the GBNC-full to the size of the JSM corpus would be too coarse to answer these questions. So our sample contains a million 1-grams and nearly 64 million 5-grams.

Preprocessing

We filter ngrams that contain special characters, e.g., figures. As definition of "allowed character", we use function `isLetter()` from the Java class `java.lang.Character`. The GBNC differentiates between different parts-of-speech of a word. Since we do not need these part-of-speech tags, we filter tagged words and only use the untagged ngrams.

10.2.2 Experimental Setup

We run our experiments on an Intel® Xeon® CPU E5-2630 v3 @ 2.40GHz. The machine has 125 GB of RAM and Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-98-generic x86_64) as operating system. To compare different technologies, we have exemplarily chosen PostgreSQL, a state-of-the-art RDBMS, and Apache Flink, a state-of-the-art MapReduce framework. With index support, RDBMSs tend to have a very good selection performance. MapReduce in turn facilitates scalable parallelization of queries.

Apache Flink

Apache Flink[5] is a distributed processing engine for streams and batch jobs. For our evaluation, we use version 1.3.2. We store the corpus in a compressed file using Kryo's JavaSerializer[6] that is shipped with Flink. Our file, containing 4.5 million 1-grams, requires 670 MB disk space.

PostgreSQL

PostgreSQL[7] is an open-source object-relational database system. We use version 11.4. We define the text attribute *ngram* as primary key and build a trigram index (`gin_trgm_ops`) as secondary index on this attribute. Our table containing 4.5 million 1-grams requires 4,400 MB disk space.

[5] https://flink.apache.org

[6] For details, see https://ci.apache.org/projects/flink/flink-docs-release-1.3/api/java/org/apache/flink/api/java/typeutils/runtime/kryo/package-summary.html.

[7] https://www.postgresql.org

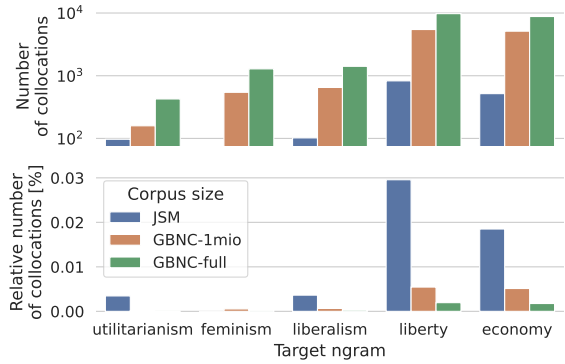

Fig. 2 A size comparison of collocation sets of words related to the research topics of John Stuart Mill over different corpus sizes.

10.3 Informative Value

To evaluate the informative value of queries, we examine the differences of results on different corpora, a small and a large one.

10.3.1 Query Template Instantiation

To compare results obtained from the three data sets, we now define customizations to instantiate our Collocation Selection Query Template. We only query words and topics related to Mill and his research topics. We select the following words.

- utilitarianism
- feminism
- liberalism
- liberty
- economy

We fix the time interval to the time domain of the JSM corpus (1963–1991) and set the radius to 5.

10.3.2 Experiments

The upper plot in Figure 2 shows the result sizes of the queries, the lower one the relative result sizes, i.e., the result size in relation to the corpus size. We see the relative result sizes as the relevance of a word within a corpus. In other words, the higher the number of collocations of a word, the more relevant it is. To provide comparable results, we normalize the number of collocations with the number of words of the corpus.

$$\text{relevance}(word) = \frac{|RCOL_{word}|}{|RC|} \qquad (12)$$

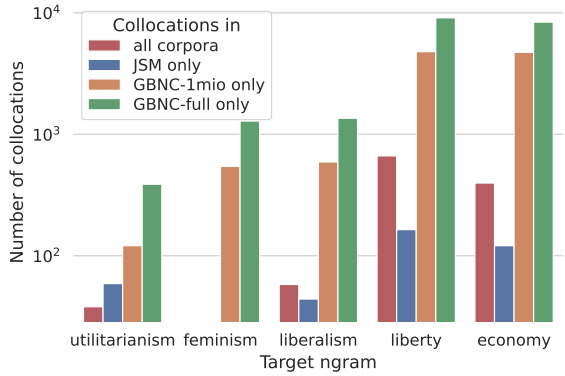

Fig. 3 A quantity comparison of collocation sets for words related to the research topics of John Stuart Mill over different corpus sizes.

Figure 3 shows the size of same and different content in the results of different corpora. To explain, please look at R_{JSM} as the result with corpus JSM and R_{GBNC} as the result with GBNC for the same query. *JSM only* are collocations that exclusively occur in the JSM corpus. We use the following abbreviations.

$$\text{all corpora} := R_{\text{all}} := R_{\text{JSM}} \cap R_{\text{GBNC-1mio}} \cap R_{\text{GBNC-full}} \tag{13}$$

$$\text{JSM only} := R_{\text{JSM}} \setminus R_{\text{all}} \tag{14}$$

$$\text{GBNC-1mio only} := R_{\text{GBNC-1mio}} \setminus R_{\text{all}} \tag{15}$$

$$\text{GBNC-full only} := R_{\text{GBNC-full}} \setminus R_{\text{all}} \tag{16}$$

10.3.3 Interpretation

We now answer the questions raised earlier.

Objective: Result Sizes

Figure 2 shows that the result size in general depends on the corpus size. Specifically, the results indicate the following: The larger the corpus, the larger is the result. The relationship, however, is not linear. The result size grows much slower than the corpus size. In other words, the relative size of the results goes down. We take this as an indication that distant reading of large corpora may be feasible in principle.

Objective: Comparison of Corpora

In our experiment, the word "liberty" has the largest collocation set on all three corpora, "economy" the second largest etc. Figure 3 shows this result. We conclude that these topics may have the same relevance in world literature as in Mill's writings. Regarding Mill's research topics, we did not find any sign that studies based on small corpora are immediately affected by filter bubbles. We also did not find any sign that preliminary investigations based on small corpora or samples yield inaccurate estimations of the true results.

This now begs to study these issue in a temporally differentiated fashion, i.e., whether the topics "behave" differently in the corpora at different times. However, this goes beyond this current evaluation and is part of future work.

Objective: Insights Regarding Content

As one might have expected, we did observe differences when comparing large corpora with smaller, more specific corpora. In general, few collocations exist only in the JSM corpus, but not in GBNC. Most collocations also occur in the GBNC.

We now examine the collocations for the words "feminism" and "utilitarianism". The JSM corpus does not contain any collocations for "feminism" beyond the ones of the GBNC. Mill was one of the first researchers publicly striving for women's rights. Today, the discussion of women's rights has evolved and arrived in society. We assume world literature to reflect this. Another research topic, which is less widespread, is "utilitarianism". Among others, we found the following collocations in the JSM corpus that are not present in GBNC: "clerical", "mysticism", and "scepticism". To our knowledge, these words relate to Mill's research contents. We conclude that world literature mainly contains general content that is mainstream in nature. To analyze specific content, one also needs a specific corpus. All in all, we conclude that our benchmark indeed provides some insights regarding the content of a corpus.

10.4 Run-Time Performance

We now benchmark the run time to execute our query templates on a RDBMS and a MapReduce framework.

10.4.1 Experiments

We run all benchmark queries 10 times on both systems and measure their execution times, from sending the query to receiving the entire result.

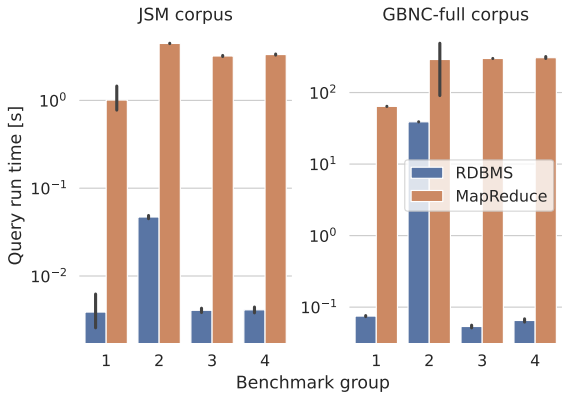

Fig. 4 A comparison of the run times between a RDBMS and a MapReduce framework to process typical queries from conceptual history using our benchmark.

Figure 4 shows the accumulated run times per benchmark group. We conclude that the selection performance is more important for short run times. This result is very plausible, since the evaluation of analytical queries also comprises the evaluation of selection subqueries at least once.

10.4.2 Summary

We have evaluated the usefulness of our benchmark to assess the performance of distant reading systems. Our evaluation shows that the different templates incur different run times on different technologies. This should enable researchers to find performance bottlenecks with our benchmark.

11 Conclusions

In the last years, the idea of distant reading has become popular, i.e., computational analyzes of large volumes of text. To compare and optimize respective systems, one needs a benchmark that helps to design and implement functionality that assists conceptual historians with their work. In this article, we have proposed a generic benchmark for distant reading. It mimics examinations of the historical semantics of words, similar to how conceptual historians actually work. Here, 'generic' means that one can apply our benchmark on arbitrary data sets. To define our benchmark, we have analyzed and formalized how conceptual historians work as well as the information they are interested in. Our benchmark enables content-related insights into a corpus as well as performance evaluations of distant reading systems.

Future Work

Given our generic benchmark for distant reading, we see various directions for future work. Three important ones are as follows. One is to extend the operations to compare collocation sets. In Section 8.2, we use intersection, union, and minus. But there are more complex operations that compare the weights of the collocations [30], e.g., with log odds ratio. A question of interest is what the user can conclude from the output of a specific operation. Another direction is to study the content-specific differences of text corpora built from different media and publication types. This will answer the question how concepts are used across media types and forms of publication. A third direction is to define and benchmark approximate operators for distant reading systems. An approximate operator is one that generates an approximation of the exact result but requires a substantially shorter execution time than its exact counterpart. Our benchmark would allow to evaluate such operators regarding both run-time performance and content-wise.

Acknowledgments. We thank Christoph Schmidt-Petri and Michael Schefczyk from the department of Philosophy of KIT for much help regarding various aspects of this work.

Declarations

Funding

This work was supported by the Ministry of Science, Research and the Arts Baden-Württemberg, project Algorithm Engineering for the Scalability Challenge (AESC).

Conflict of interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

[1] Abiteboul S, Hull R, Vianu V (1995) Foundations of Databases: The Logical Level. Addison-Wesley, Reading, Mass

[2] Bakshy E, Messing S, Adamic L (2015) Exposure to ideologically diverse news and opinion

on facebook. Science 348(6239):1130–1132. https://doi.org/10.1126/science.aaa1160

[3] Barnbrook G, Mason O, Krishnamurthy R (2013) Collocation and language theory: recent developments. In: Collocation. Palgrave Macmillan UK, p 147–173, https://doi.org/10.1057/9781137297242_7

[4] Blank A (2012) Prinzipien des lexikalischen Bedeutungswandels am Beispiel der romanischen Sprachen. De Gruyter

[5] Burrows S, Falk M (2021) Digital humanities. https://doi.org/10.1093/acrefore/9780190201098.013.971

[6] Deerwester S, Dumais S, Furnas G, et al (1990) Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6):391–407. https://doi.org/10.1002/(sici)1097-4571(199009)41:6⟨391::aid-asi1⟩3.0.co;2-9

[7] Devlin J, Chang MW, Lee K, et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. CoRR https://arxiv.org/abs/1810.04805

[8] Elekes A, Schäler M, Böhm K (2017) On the various semantics of similarity in word embedding models. In: ACM/IEEE Joint Conference on Digital Libraries (JCDL). IEEE, pp 1–10, https://doi.org/10.1109/jcdl.2017.7991568

[9] Englhardt A, Willkomm J, Schäler M, et al (2019) Improving semantic change analysis by combining word embeddings and word frequencies. International Journal on Digital Libraries 21(3):247–264. https://doi.org/10.1007/s00799-019-00271-6

[10] Firth J (1957) A synopsis of linguistic theory, 1930-1955. Studies in linguistic analysis

[11] Flaxman S, Goel S, Rao J (2016) Filter bubbles, echo chambers, and online news consumption. Public Opinion Quarterly 80(S1):298–320. https://doi.org/10.1093/poq/nfw006

[12] Foxlee N (2015) From analogue to digital: Conventional and computationalapproaches to studying conceptual change. In: Conceptual change: Digital Humanities Case Studies

[13] Friedrich A, Biemann C (2016) Digitale begriffsgeschichte? methodologische Überlegungen und exemplarische versuche am beispiel moderner netzsemantik. Forum Interdisziplinäre Begriffsgeschichte (FIB)

[14] Fritz G (2006) Historische Semantik. J.B. Metzler, https://doi.org/10.1007/978-3-476-01408-5

[15] Fritz G (2011) Einführung in die historische Semantik. De Gruyter

[16] Hamilton W, Leskovec J, Jurafsky D (2016) Cultural shift or linguistic drift? comparing two computational measures of semantic change. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, https://doi.org/10.18653/v1/d16-1229

[17] Hamilton W, Leskovec J, Jurafsky D (2016) Diachronic word embeddings reveal statistical laws of semantic change. CoRR https://arxiv.org/abs/1605.09096

[18] Heringer H (1999) Das höchste der Gefühle: Empirische Studien zur distributiven Semantik. Stauffenburg Verlag

[19] Kistowski J, Arnold J, Huppler K, et al (2015) How to build a benchmark. In: Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering. ACM, pp 333–336, https://doi.org/10.1145/2668930.2688819

[20] Koselleck R (2004) Futures Past: On the Semantics of Historical Time (Studies in Contemporary German Social Thought.). Columbia University Press

[21] Koselleck R (2006) Begriffsgeschichten: Studien zur Semantik und Pragmatik der politischen und sozialen Sprache. Suhrkamp Verlag, Frankfurt am Main

[22] Levy O, Goldberg Y (2014) Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, vol 27. Curran Associates, Inc., pp 2177–2185

[23] Levy O, Goldberg Y, Dagan I (2015) Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics 3:211–225. https://doi.org/10.1162/tacl_a_00134

[24] Lin Y, Michel JB, Aiden E, et al (2012) Syntactic annotations for the google books ngram corpus. In: Proceedings of the ACL 2012 System Demonstrations. Association for Computational Linguistics, USA, pp 169–174

[25] Macleod C (2018) John stuart mill. In: Zalta E (ed) The Stanford Encyclopedia of Philosophy, fall 2018 edn. Metaphysics Research Lab, Stanford University

[26] Maier D (1983) Theory of Relational Databases. Computer Science Press, Rockville, Md

[27] Manovich L (2016) The science of culture? social computing, digital humanities and cultural analytics. Journal of Cultural Analytics 1(1). https://doi.org/10.22148/16.004

[28] Michel JB, Shen Y, Aiden A, et al (2011) Quantitative analysis of culture using millions of digitized books. Science 331(6014):176–182. https://doi.org/10.1126/science.1199644

[29] Mikolov T, Sutskever I, Chen K, et al (2013) Distributed representations of words and phrases and their compositionality. In: Burges C, Bottou L, Welling M, et al (eds) Advances in Neural Information Processing Systems, vol 26. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[30] Monroe B, Colaresi M, Quinn K (2008) Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. Political Analysis 16(4):372–403. https://doi.org/10.1093/pan/mpn018

[31] Moretti F (2016) Distant Reading. Konstanz University Press

[32] Naber D (2005) Openthesaurus: Ein offenes deutsches wortnetz. In: Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen: Beiträge zur GLDV-Tagung 2005 in Bonn, pp 422–433

[33] Organisciak P, Capitanu B, Underwood T, et al (2017) Access to billions of pages for large-scale text analysis. In: iConference 2017 Proceedings, vol 2. iSchools, pp 66–76, URL http://hdl.handle.net/2142/98873

[34] O'Connor M, Das A (2009) Sqwrl: A query language for owl. In: Proceedings of the 6th International Conference on OWL: Experiences and Directions, OWLED'09, vol 529. CEUR-WS.org, pp 208–215

[35] Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)

[36] Prabhakaran V, Hamilton W, McFarland D, et al (2016) Predicting the rise and fall of scientific topics from trends in their rhetorical framing. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, https://doi.org/10.18653/v1/p16-1111

[37] Rosario B (2000) Latent semantic indexing: An overview. In: INFOSYS 240

[38] Saussure F (1998) Course in General Linguistics. Open Court, LaSalle, Ill

[39] Snodgrass R (1987) The temporal query language TQuel. ACM Transactions on Database Systems 12(2):247–298. https://doi.org/10.1145/22952.22956

[40] Snodgrass R (1995) The TSQL2 Temporal Query Language. The International Series in Engineering and Computer Science, Springer US

[41] Spohr D (2017) Fake news and ideological polarization: Filter bubbles and selective exposure on social media. Business Information Review 34(3):150–160. https://doi.org/10.1177/0266382117722446

[42] Steyer K (ed) (2004) Wortverbindungen - mehr oder weniger fest. De Gruyter, https://doi.org/10.1515/9783110622768

[43] Subramanian S, King D, Downey D, et al (2021) S2and: A benchmark and evaluation system for author name disambiguation. In: 2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL). IEEE, pp 170–179, https://doi.org/10.1109/jcdl52503.2021.00029

[44] Wahle J, Ruas T, Meuschke N, et al (2021) Are neural language models good plagiarists? a benchmark for neural paraphrase detection. In: 2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL). IEEE, pp 226–229, https://doi.org/10.1109/jcdl52503.2021.00065

[45] Webster J, Watson R (2002) Analyzing the past to prepare for the future: Writing a literature review. MIS Quarterly 26(2):xiii–xxiii

[46] Willkomm J, Schmidt-Petri C, Schäler M, et al (2018) A query algebra for temporal text corpora. In: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries. ACM, https://doi.org/10.1145/3197026.3197044

[47] Wolf M, Horn A, Mehl M, et al (2008) Computergestützte quantitative textanalyse. Diagnostica 54(2):85–98. https://doi.org/10.1026/0012-1924.54.2.85