Low-cost Semantic Enhancement to Digital Library Metadata and Indexing: Simple Yet Effective Strategies

Annika Hinze
University of Waikato
Hamilton, New Zealand
hinze@waikato.ac.nz

David Bainbridge
University of Waikato
Hamilton, New Zealand
davidb@waikato.ac.nz

Sally Jo Cunningham
University of Waikato
Hamilton, New Zealand
sallyjo@waikato.ac.nz

J. Stephen Downie
Library & Information Science
University of Illinois
jdownie@illinois.edu

ABSTRACT

Most existing digital libraries use traditional lexically-based retrieval techniques. For established systems, completely replacing, or even making significant changes to the document retrieval mechanism (document analysis, indexing strategy, query processing and query interface) would require major technological effort, and would most likely be disruptive. In this paper, we describe ways to use the results of semantic analysis and disambiguation, while retaining an existing keyword-based search and lexicographic index. We engineer this so the output of semantic analysis (performed off-line) is suitable for import directly into existing digital library metadata and index structures, and thus incorporated without the need for architecture modifications.

CCS Concepts

•**Computing methodologies** → **Semantic networks;** *Lexical semantics;* •**Applied computing** → **Digital libraries and archives;** •**Information systems** → *Digital libraries and archives; Search engine indexing;*

Keywords

Semantic analysis; disambiguation; indexing; semantic enrichment

1. INTRODUCTION

Search in large collections—such as the bespoke solutions developed for the HathiTrust Digital Library (HTDL, www.hathitrust.org) and Google Books (books.google.com) or those built through general purpose digital library software, such the Greenstone toolkit (www.greenstone.org)—is at the core of the services provided by a digital library. Most of these established systems provide access primarily by string-based search over inverted indexes [8] of both document full-texts and metadata, with text-based search that is implemented using lexicographic analysis (such as Solr/Lucene indexes).

JCDL '16, June 19-23, 2016, Newark, NJ, USA

DOI: http://dx.doi.org/10.1145/2910896.2910910

Most scholars using these digital libraries, however, are interested not in simple textual keywords but rather in semantic concepts. Having to express their search as keywords is restrictive in its expressiveness, and of limited use for exploring whole collections. Lexicographic search in large collections, such as the HathiTrust's 13,000,000 volumes with 4.6 billion pages, often returns large sets of unrelated documents (due to homographs —same spelling, different meaning—being included), while relevant sources may remain undetected unless the right keyword is found. The problem is exacerbated in documents that have been obtained through Optical Character Recognition (OCR), as recognition errors may lead to misidentification of terms, which are then either mistakenly included or omitted from the search results.

In the Capisco project [3], we introduced a new way of semantic search in large collections that affords the benefits of semantic search while minimizing the problems associated with applying existing semantic analysis at scale. The developed software architecture avoids the need for complete semantic document markup using pre-existing ontologies by developing an automatically generated *Concept-in-Context* (*CiC*) network seeded by *a priori* analysis of Wikipedia texts and identification of semantic metadata. Capisco also provides the means to manually introduce or modify concepts in this *CiC* knowledge base. The disambiguation of large document collections is done automatically using the *CiC* network as a knowledge base for semantic analysis. Capisco's search interface guides the user through a manual disambiguation of query terms into semantic concepts. We showed in [3] that using Capisco reduces the number of *false positives*, includes documents missed when using keyword-based search (increased *true positives*), and can to some extent remedy OCR problems (by excluding out-of-context concepts, which may stem from OCR errors).

To use Capisco in already established digital library systems, however, would require making major changes to the document retrieval mechanism, such as the introduction of semantic document analysis, and changing the indexing strategy, query processing and query interface. Such a major technological change would most likely be disruptive for both the digital library users and the software maintenance team. In this paper, we explore ways of extending the conventional digital library's retrieval capabilities by importing the results of semantic analysis and disambiguation (such as those from Capisco), while retaining an existing keyword-based search and lexicographic index.

The remainder of the paper is structured as follows: Section 2 briefly describes related work on using concepts and semantic relationships mined from Wikipedia. Section 3 outlines the basic com-

ponents of typical lexical indexing in Digital Libraries (DLs) and indexing in Capisco with specific focus on semantic disambiguation and indexing structures. Section 4 then discusses options to incorporate semantic information into existing DL systems. Section 5 presents results for a number of small test collections, and compares the quality of the results. We then show a worked example of a system using semantic enhancements to index and metadata in Section 6. Section 7 discusses related approaches, lessons-learned from our explorations, and future research directions, while Section 8 concludes the paper.

2. RELATED WORK

Our research is situated in the body of work exploring the application to information retrieval of concepts and semantic relationships mined from Wikipedia. The key insight in this use of Wikipedia is to treat its structure (rather than the contents of its articles) as a semantic resource: the title for each article is a brief phrase describing a single concept, and the links between articles capture the hierarchical and associative relationships between concepts. "Redirects" link alternative expressions of a concept (e.g., synonyms, abbreviations, spelling variations, colloquialisms, scientific terms, *etc.*) to the concept term (the article title) [10].

The mined Wikipedia structures have been exploited to enhance different aspects of a retrieval system architecture and to support user searching: to automatically create a domain-specific thesaurus [10] or a general thesaurus covering all Wikipedia topics [10, 13], to cluster documents based on the semantic relatedness of their associated Wikipedia concepts/articles [4], and to develop search interfaces that support semantic-based query expansion and query refinement [12, 3]. Key to all of these applications is efficient support for automatically cross-referencing document terms with their associated Wikipedia link structure [11, 9].

This earlier work largely developed proof-of-concept systems and included evaluations to establish the potential *effectiveness* of mined concepts and relationships for improving search. In contrast, in this present paper we investigate the *practicality* of including mined concepts and relationships into existing DLs. Our goal here is to explore the opportunities to gain the benefits of semantic enhancements while requiring minimal, if any, changes to a DL's underlying architecture and interface.

3. BACKGROUND: LEXICAL AND SEMANTIC INDEXING

In this section, we introduce an example user's information need and a set of documents that will be used to explore the implications of our different search strategies. We then outline the indexing processes, data structures and search options for both lexical indexing (typical for DLs) and semantic indexing (Capisco), in Sections 3.2 and 3.3, respectively. We assume the existence of a collection of documents, in which each of the documents consists of one page or more and the length of pages is variable. Figure 1 shows a simplified structure of a document (here consisting of three pages) and the catalogue's bibliographic metadata for the document. We focus on text-based indexing; similar processes and structures would be used for other media types. For simplicity, we use *term* to refer to single words, phrases, or n-grams (sequence of n words).

3.1 Example query and documents

Throughout the paper, we use the main example of a user querying for information about the Pacific island nation of Niue over a collection of documents. We selected four example documents to highlight the implications of the various indexing and search strategies, see Table 1.[1] The documents are referred to as D1–D4 with qualifiers indicating whether a document is relevant for our user's information need regarding the Niue island (superscript $+$ or $-$) and a shorthand notation of the query relevant OCR-ed term in the document (subscript).

Doc-ID	OCR-ed term	Semantics	Match
$D1^{+}_{\text{Niue}}$	"Niue"	*Niue*	yes
$D2^{+}_{\text{SI}}$	"Savage Island"	*Niue*	yes
$D3^{-}_{\text{SI}}$	"Savage Island"	*primitive place*	no
$D4^{-}_{\text{Niue}}$	"Niue"	*Nine*	no

Table 1: Example documents for information need *Niue*

$D1^{+}_{\textbf{Niue}}$ The first document is a 1970 book about the *Flora of Niue* [15]. It contains the literal term "Niue" both in its title and on several pages throughout the book. The document is relevant to the user's information need.

$D2^{+}_{\textbf{SI}}$ The second document is an historic collection from 1889 of reports about Niue and other Pacific islands [1]; it is also relevant to the user's information need. As was customary at the time, it refers to the Niue island by the term "Savage Island".

$D3^{-}_{\textbf{SI}}$ The third document is John Redmond's 1910 speech to the Irish Parliament, in which he refers to the Irish Railway system of the time as being as "neglected as if it had been a savage island in some distant ocean" [14]; it is not relevant to the user's information need.

$D4^{-}_{\textbf{Niue}}$ Finally, the fourth document is an historic treatise published 1870 about the church in Wales [5]. It contains an OCR error that interprets the word "Nine" as "Niue". This book is also not relevant to the user's information need.

We now explain how both lexical full-text search and semantic search using Capisco execute this example search and the implications for the respective result sets. This explanation forms the foundation for exploring (in Section 4) how to merge Capisco semantic data into existing digital library metadata and index structures without the need for architecture changes.

3.2 Lexical indexing

Typical full-text indexing (e.g., as provided by Solr[2]) analyzes the contents of each text page (performing lexical transforms such as case folding, stop-word removal and stemming) and creates for each term an index entry with references to the pages on which the term appears (see Figure 1, top). The bibliographic metadata is typically kept in a separate structure, where each metadata field (such as author, title, subject) carries one or several entries, which link to the document (see Figure 1, bottom). Some DL implementations additionally include the metadata in the full-text index.

Common search interfaces for DLs offer 'simple full-text search' and 'advanced full-text search'. Simple search typically offers a single query box for keywords or phrases (see mock-up in Figure 2, left) and executes a search via the full-text and all metadata fields. The advanced search option typically allows a user to specify a Boolean combination of searches in full-text and in each of the metadata fields (see Figure 2, top right). Additionally some

[1]These documents and other test collections have been provided by the HathiTrust.

[2]lucene.apache.org/solr/

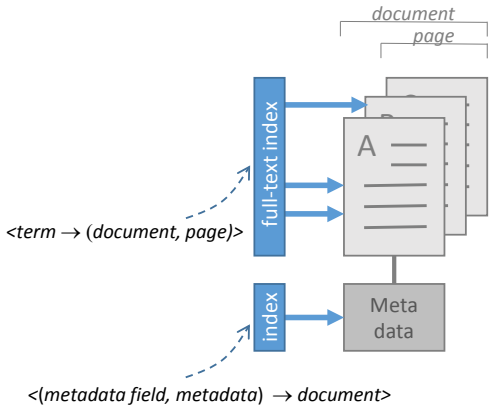

Figure 1: Lexicographic indexing

systems support filtering of results by selected metadata fields (see Figure 2, bottom right). The only metadata fields that can be used for filtering are those that have numerous entries (such as language or format) out of which the user can select the appropriate ones.

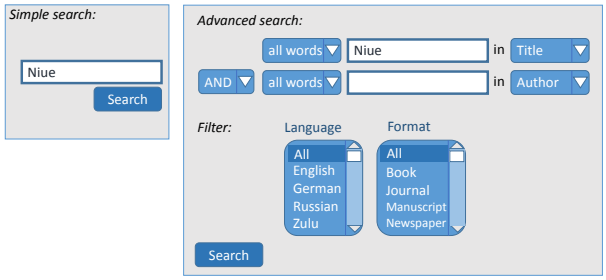

Figure 2: Mock-up of Simple (left) and Advanced (right) Lexical Search interfaces for DL collections

Example 1. (Lexical index & search) *Let us now consider our four example documents $D1$ to $D4^{-}_{Niue}$ introduced above. Figure 3 shows a simplified full-text index structure (as used in Fig. 1, top) for the terms "Niue" and "Savage Island" appearing in each of the four documents.*[3] *Even though the two terms have different semantic meanings in the documents, each occurrence will be treated the same in the index.*

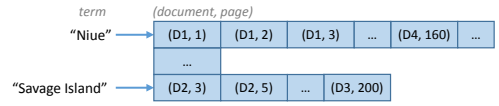

Figure 3: Example index structure for full-text (snippet)

Using this index in simple full-text search (see Figure 2, left) will result in documents $D1^{+}_{Niue}$ and $D4^{-}_{Niue}$ being included in the result list, and $D2^{+}_{SI}$ and $D3^{-}_{SI}$ being omitted. $D4^{-}_{Niue}$ is a false positive and $D2^{+}_{SI}$ is a false negative.

Using advanced search (see Figure 2, upper right) on the bibliographic metadata field title, alone or in conjunction to the full-text search, yields the result set $D1^{+}_{Niue}$ (removing the false positive $D4^{-}_{Niue}$). This approach, however, excludes other matching documents that do not carry to term "Niue" in their title (such as $D2^{+}_{SI}$). Additional filtering of bibliographic metadata by language or format (see Figure 2, lower right) is not useful here as all four documents are books in English.

[3]For simplicity, we abstract from the precise locations in which the terms appear on each page.

3.3 Capisco: Semantic indexing

Capisco uses a knowledge base containing information about concepts in context, initially created by mining Wikipedia and potentially further enriched by domain experts (see orange elements in Fig. 7). Each concept is identified by an *id*, and also carries a human-readable *concept label*. Concept labels are derived from Wikipedia article titles. Synonymous terms for a concept are stored with reference to the context in which they appear. The context of a term refers to the main area in which this term is used for this concept (e.g., term "Apple" refers to concept *Apple Record* in the context of *music* and to the concept *Malus domestica* in the context of *horticulture*). Because contexts are also concepts, the knowledge base forms an interlinked Concepts in Context (CiC) network.

Capisco processes each document by first disambiguating each term (i.e., identifying its semantic concept) by reverse look-up of the term in the knowledge base (i.e., querying all synonyms) to identify potentially matching concepts. These concepts are then disambiguated by filtering out those concepts for which no valid context can be found on the document page. This leads to the identification of *significant* topics within a document (i.e., not every noun found in the document matches a concept). For further detail on the disambiguation process, see [3]. The identified concepts for each document are then indexed into a full-concept index, analogous to traditional full-text indexes. For each context, an index entry is created with references to the pages on which the term appears (see Figure 4, middle). The index carries only concept IDs; concept labels and synonyms can be accessed via the knowledge base (see Figure 4, left). Due to Capisco's focus on semantic search, it does not currently employ an index for the metadata; if it did, the index would be identical to the one in lexical indexing.

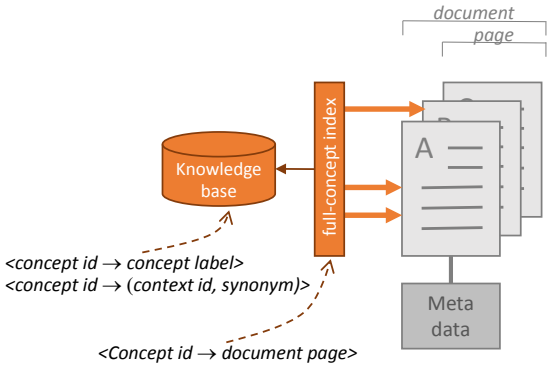

Figure 4: Capisco indexing

Capisco's search interface starts similarly to lexical search by the user inserting search terms (see mock-up in Figure 5, left). One or more keywords can be entered; our example shows the process for the single term "Puck". For each search term, Capisco retrieves all potentially matching concepts and presents these to the user for manual disambiguation (Figure 5, right). Once the user has selected the concept that matches their search interest, the search process is executed via a lookup of the concept IDs in the full-context index. A detailed description of the process with a walk-through with interface screen-shots can be found in [3].

Example 2. (Semantic index & search) *Figure 6 shows the full-concept index structure in Capisco for the terms "Niue" and*

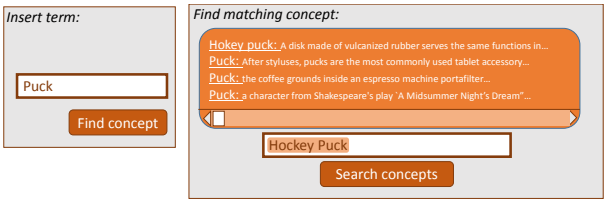

Figure 5: Mock-up of Manual Disambiguation and Search interface

"Savage Island" appearing in our four documents. The terms that are used synonymously for Niue Island (which has the concept ID 14334*) are referred to in the knowledge base, while the index itself only contains the concept ID. Note that the use of the phrase "savage island" in document* $D3^{-}_{SI}$ *is not related to a specific concept, as it was merely a figure of speech, and is therefore not included in the semantic index. Neither is the term "Niue" from document* $D4^{-}_{Niue}$ *included, as there is no context supporting such a semantic reading.*

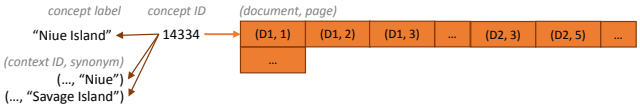

Figure 6: Example index structure for Capsico (snippet)

Searching using Capisco, the user is first guided through the manual disambiguation of the search term "Niue" to the concept Niue Island *(id* 14334*), using an interface like the one shown in Figure 5. Capisco then uses the full-concept index (see Figure 6) to look-up the documents for ID=*14334*. This will result in documents* $D1^{+}_{Niue}$ *and* $D2^{+}_{SI}$ *being included in the result list, and* $D3^{-}_{SI}$ *and* $D4^{-}_{Niue}$ *being omitted. In this case, a concept search would return no false negatives or false positives.*

4. ENHANCING LEXICAL SEARCH THROUGH SEMANTICS

We now explore strategies that allow us to use the semantic information gained by employing Capisco (see orange elements in Fig. 7), without completely replacing the established lexicographic document analysis, indexing technique, or query interface (see blue elements in Fig. 7). We explore four options open to standard digital libraries (Sections 4.1–4.4), and one further option for DLs supporting advanced metadata structures (Section 4.5). We will explore the complexity of the solution (in terms of changes needed in the DL system's interface and indexing) and implications for query formulation and result quality (by exploring the potential for introducing false negatives and false positives into search results).

4.1 Concept labels added to metadata

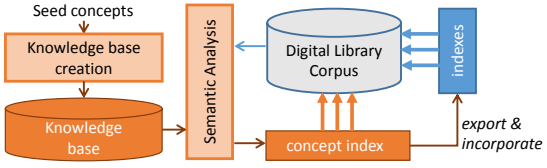

Figure 7: Export of semantic concepts into existing DL indexes

The most straight-forward way to incorporate the concepts that have been identified for each document is to create a new metadata field at the document level. Each of the concept labels is then automatically added to this concept field, similar to the treatment of entries in a subject metadata field. Fig. 8 illustrates this approach.

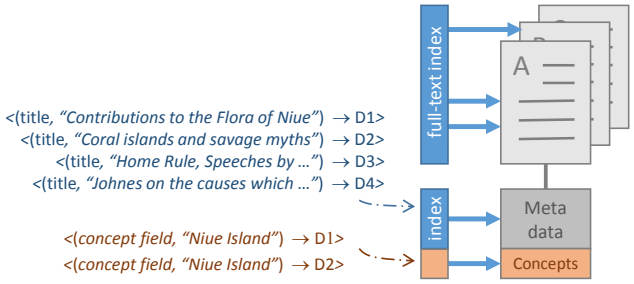

Figure 8: *Approach 1*–Concept labels added to metadata; extended indexing structure (right) and example snippet (left)

Searching for a document using a digital library's advanced catalogue search may then use this metadata field as one of the options either in a search or else as a filter applied at the time of searching. An example of filtering can been seen in the HathiTrust advance query page[4] where the union of all the languages the collection is written in is displayed as a list: selecting an item from the list restricts the search to texts written in that language. In our case of a concept added to the metadata, using this field as a conjunctive search option restricts the result set to those documents for which a matching concept label is found. The success of this strategy, therefore, depends on the user choosing the exact same search keyword as appears in the matching concept label.

Example 3. Approach 1–search *Using this indexing approach with our four documents leads to concept labels being added to the metadata as indicated in Figure 8, left. Results for both simple and advanced search now depend on whether the DL indexer uses a tokenizer: either the metadata is interpreted as the phrase "Niue Island" or interpreted as two words. In the first case, a simple or advanced search for "Niue" will lead to the same results as pure lexical search (see Example 1). In the second case, executing a simple search for "Niue" evaluates the full-text and all metadata fields, leading to a result set containing* $D1^{+}_{Niue}$, $D2^{+}_{SI}$ *and* $D4^{-}_{Niue}$*. When comparing to the purely lexical search result (see Example 1), we note that the false positive* $D4^{-}_{Niue}$ *is retained while the false negative* $D2^{+}_{SI}$ *from lexical search is remedied. The result of the advanced search for "Niue" in the concept field alone leads to* $D1^{+}_{Niue}$ *and* $D2^{+}_{SI}$*, thus removing the false positive* $D4^{-}_{Niue}$ *and addressing the false negative* $D2^{+}_{SI}$ *from lexical search. We note that the user here selected a matching keyword "Niue"; any other keyword, such as "Niu" or "Niue-Fekai" (both alternative names for the island) are unsuccessful.*

Using advanced search with filtering (such as shown in Figure 2 bottom right) may have better outcomes. Each concept is represented by one term only and users would select one such concept; thus metadata filtering offers greater transparency to the user (than in metadata search). However, users are still required to scan the complete (alphabetically ordered) list. If the concept labels are listed without further information about their semantic meaning,

[4]Such as the advanced search for HathiTrust items at catalog.hathitrust.org/Search/Advanced

misunderstandings are hard to avoid. Providing additional semantic information about a concept would therefore be advised. Overall concept metadata filtering is a simple and easy to achieve option, tempered with the observation that it has some challenges for search interface usability.

If the user filters by the correct concept, the search results will filter out those false positives that would be included in the ordinary lexicographic search through homonyms. However, if they combine the filter with a non-matching full-text search term, the result set is reduced (different to semantic search in Capisco). Best results are therefore achieved if the search keyword matches the concept filter (thus semantically narrowing the search for the term).

Finally, the result list may include large numbers of correctly matching documents which are nevertheless not of particular interest (i.e., no ranking by concept filter). This could be offset through only including the n most important concepts (although this in turn would then limit the filter capability).

Example 4. Approach 1–filter *To use a filter on the concept field, our user first needs to identify a suitable concept label from a list. Fortunately, in this case the concept label "Niue island" is similar to the user's search term "Niue". Filtering by concept field only, leads to the results* $D1^{+}_{Niue}$ *and* $D2^{+}_{SI}$ *(again removing the false positive* $D4^{-}_{Niue}$*, and omitting the negative* $D2^{+}_{SI}$ *from lexical search). Using an advanced search, e.g., for "Niue", on the full-text in conjunction with the filter by concept field* Niue Island*, reduces the result set to* $D1^{+}_{Niue}$ *(with false negative* $D2^{+}_{SI}$*). Users may habitually try to ask for their search term in several fields, and may therefore not be familiar with these implications.*

4.2 Concepts & synonyms added to metadata

Approach 1, above, is limited by using the concept label only in the metadata. This second approach aims to remedy this by including not only concepts but also all concept synonyms into the same metadata field (see Figure 9 for illustration).

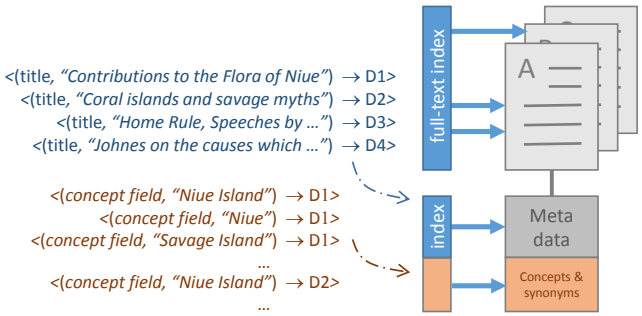

Figure 9: *Approach 2*–Concepts and synonyms added to metadata; extended indexing structure (right) and example snippet (left)

An advanced search using keywords in the concept metadata field will yield much better results than for Approach 1, because the keywords have a much higher probability to be matched by one of the concept synonyms. The best strategy would be to filter by (or search for) a concept and all its synonyms and also include these as keywords in the full-text search.

Example 5. Approach 2–search *Using this indexing approach with our four documents means that the concept label and synonyms will be added as metadata (see Figure 9, left). In this approach, the different indexing options for the metadata fields (phrases vs single words) does not lead to different results. A simple query for "Niue" would, different to Example 3, lead to* $D1^{+}_{Niue}$*,* $D2^{+}_{SI}$ *and* $D4^{-}_{Niue}$*, because several synonymous terms for the concept* Niue *are available.*

Advanced search for "Niue" in the concept field *alone would lead to a result set containing* $D1^{+}_{Niue}$ *and* $D3^{-}_{SI}$*, thus removing the false positive* $D4^{-}_{Niue}$ *and omitting the negative* $D2^{+}_{SI}$ *from lexical search. Any other keywords, such as "Niu" or "Niue-Fekai" lead the the same result.*

The advanced search strategy using filters would be similar to the one described for Approach 1, with the difference that now the potential list of concepts and their synonyms is much longer. However, a user would be more likely to be able to identify the concept they have in mind (as each concept now has several expressions instead of just one).

The same limitations with regard to the lack of ranking by concept applies as in the Approach 1. Overall, though, the quality of the search results is expected to be better. The number of false positives is potentially reduced (compared to Approach 1) because the user has more opportunity to identify their desired concept (via filter or search). Those search results that would be mistakenly excluded in lexicographic search (false negatives) because a keyword is not matched, could now be included by searching for the keyword in the concept field, which would lead to better results due to the inclusion of synonyms. Here a new set of false positives, however, could now be included as these synonyms are no longer linked to their semantic concept, nor is the semantics of the search keyword clearly identified. The problem of filter results including false positives is increased as more potential matches are now included. Overall this approach offers more options and better user support, but its effectiveness may depend heavily on the collection.

Example 6. Approach 2–filter *When filtering by* concept field *our user now has several options to identify a matching metadata entry (e.g., Niue Island, Niue, Niu, Niue-Fekai). Filtering by concept field only leads to the results* $D1^{+}_{Niue}$ *and* $D2^{+}_{SI}$*. Using an advanced search, e.g., for "Niue", on the full-text in conjunction with the filter by concept again leads to* $D1^{+}_{Niue}$ *only (with false negative* $D2^{+}_{SI}$*).*

4.3 Concept labels indexed at page-level

In contrast to the first two approaches, Approach 3 does not merely change metadata but includes the concepts into the index for each page. This means that each page is (virtually) extended by a number of concept keywords, which are in the indexing process being treated as part of the page content. Figure 10 illustrates the approach.

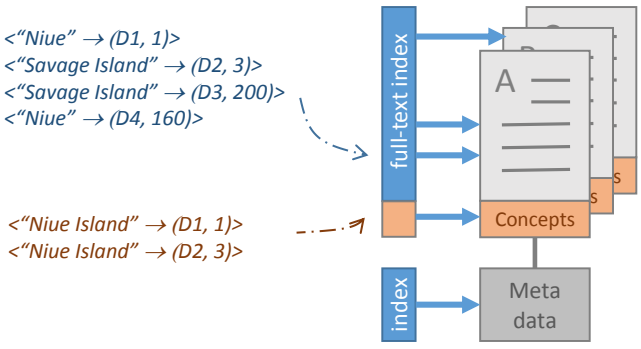

Figure 10: *Approach 3*–Concept labels indexed at page-level; extended indexing structure (right) and example snippet (left)

No change in the user's search interface is required, as the enhancement happens at the indexing level. The chances of semantically matching documents being found for a search are increased, as all concept synonyms in a text are additionally tagged by the concept label. That is, if the document contains a lesser-known term, while the user searches for the more widely used term (which, we assume, matches the concept label) then the document is retrieved. In purely lexical search, the user would have missed this result (false negative). The number of false positives is not directly affected, as there is no filtering (as in Approaches 1 and 2) and potentially more terms can be matched. However, because the context label is included as a term (without semantic meaning), it is now susceptible to being 'misunderstood' during the lexical search (i.e., being a homonym to an unrelated concept), and to introduce new false positives.

Overall, the number of these should be relatively small, as the concept labels are largely very specific, but they can occur. In some documents, the addition of the concept labels may lead to an increased repetition of terms (i.e., if the term used in the document is identical to the concept label). In these cases, naturally there is no change to the number of false negatives (as the document was already included in the result list). However, the increased repetition of the term would lead to a higher ranking of the document within the result list. Documents that use terms other than the one matching the result label, while talking about the same concept, would not be boosted in the same way. Overall, this approach seems to favour documents using established terminology, while also retrieving those using more unusual concept terms.

Example 7. Approach 3 *Using this indexing approach with our four documents leads to concept labels being added into the full-text index (see Figure 10, left). If the concept label is not tokenized, i.e. interpreted as a phrase by the indexer, our user's full-text search for "Niue" leads to the same result as the lexical search, $D1^{+}_{Niue}$ and $D4^{-}_{Niue}$ (see Example 1), because the additionally indexed terms do not match the query. If the user were to search for "Niue Island" instead, the result would be $D1^{+}_{Niue}$ and $D3^{-}_{SI}$. If the concept label is tokenized, the result set for query "Niue" consists of $D1^{+}_{Niue}$, $D2^{+}_{SI}$ and $D4^{-}_{Niue}$. In this case, $D1^{+}_{Niue}$ receives a higher ranking as before as each occurrence of "Niue" now appears twice per page. False positive $D4^{-}_{Niue}$ is still included but ranked very low (single occurrence of "Niue", no re-enforcement through concept labels). $D2^{+}_{SI}$ would be ranked similarly low.*

4.4 Concept label and synonyms indexed

Approach 3 is somewhat limited by the restriction to include the concept label only. Approach 4 aims to overcome this limitation by including not only concept labels but also all concept synonyms into the page such that they are included in the full-text index (see Figure 11).

As before, no change in the user's search interface is required. If the search term matches on the concept synonyms, and assuming correct semantic analysis, all relevant documents should be included in the result set (accompanied by significant reduction in false negatives). At any rate, the results (with regards to false negatives) has the same quality as a semantic search using Capisco. However, because all synonyms are treated as terms only (i.e., without their semantic meaning attached), each of the terms is now open to be misunderstood as belonging to a different concept. Thus the number of false positives is potentially increased (compared to Approach 3). These might be filtered out by a combination with Approaches 2 or 3 (as discussed later in Section 7). Similar to Approach 3, all documents mentioning a concept will receive boosted

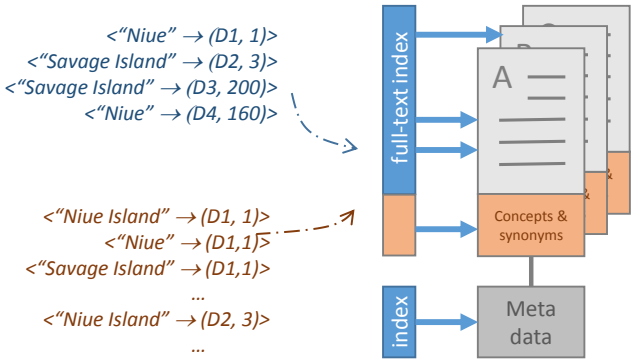

Figure 11: *Approach 4*–Concept label and synonyms indexed; extended indexing structure (right) and example snippet (left)

ranking (through the repetition of the matching concept label or synonyms), but without the skewing of results in favour of texts who use established terminology (i.e., matching the concept label).

Overall, this approach significantly reduces false negatives without users having to change their search habits. However, it may increase the number of false positives. This depends on the document collection; if a wide variety of document subjects is represented, the probability of homonyms occurring is increased. A large collection may very well contain text about both ice hockey ("hockey puck") and Shakespeare ("Puck" in Midsummer Night's Dream). In contrast, specialised collections will likely encounter this problem to a limited extent.

Example 8. Approach 4 *In this approach, concept labels and synonyms are added into the full-text index (see Figure 11, left). In this case tokenization only influences possible rankings, not the selection or exclusion of documents. The search result set for query "Niue" consists of $D1^{+}_{Niue}$, $D2^{+}_{SI}$ and $D4^{-}_{Niue}$. In this case, both $D1^{+}_{Niue}$ and $D2^{+}_{SI}$ receive higher rankings due to repetition of the term through the synonyms. False positive $D4^{-}_{Niue}$ is still included but ranked very low (single occurrence of "Niue", no re-enforcement through concept labels).*

4.5 Concepts & synonyms at page metadata

The final approach that we analyse is only possible for those DL implementations that support page-level metadata fields (such as provided in Greenstone [16]). Following Approaches 1 and 2, concept labels and concept synonyms are added as page-level metadata fields (see Figure 12 for illustration).

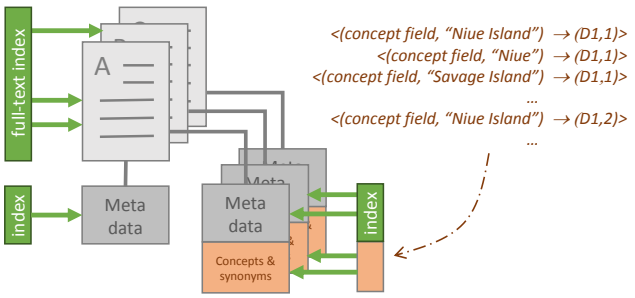

Figure 12: *Approach 5*–Concepts & synonyms at page metadata; extended indexing structure (left) and example snippet (right)

Collection	Coll. size	Tokens	Concepts	Shared	Combined size	(Combined w/synonyms)
A	701 pages	190475	2850 (12%)	783 (2.1%)	205755 (110%)	278699–369879 (149–197%)
B	878 pages	354954	45243 (13%)	13205 (3.7%)	386992 (109%)	515144–675334 (145–190%)
C	1214 pages	518401	29857 (6%)	4642 (0.9%)	543616 (105%)	644476–770551 (124–149%)
D	1649 pages	4379593	75544 (2%)	12580 (0.3%)	4442557 (101%)	4694413–5009233 (107–114%)

Table 2: Test collections and index sizes in #tokens (Solr) and #concepts (Capisco)

This metadata field may then be used in advanced search as a filter to restrict the result set, or in fielded search to increase the result set (reduction of false negatives). As in Approaches 1 and 2, the list of concepts and their synonyms may be quite large. Selecting a concept from a filter list would be easier than having to identify the correct keyword while searching the metadata field. Again, the advanced search using keywords in the concept and/or synonyms metadata fields would be useful for improving result set quality (increase in true positives in comparison to lexical search). As before, a new set of false positives may, however, now be included as these synonyms are no longer linked to their semantic concept, nor is the semantics of the search keyword clearly identified. The lack of ranking information when filtering by document metadata (as in Approaches 1 and 2) is remedied here as the frequency of pages on which a concept is used is known.

Overall the number of false positives and false negatives would be the same as in Approach 1 or 2, depending on the index variation. However, the main advantages of this approach is the support for ranking and the increased ease for a user in identifying why a document has been included in the result set, as each of the matching pages is identified.

Example 9. Approach 5 *Revisiting our four documents, the metadata changes are indicated in Figure 12, right. The search results for this approach contain the same documents as Approaches 1 and 2 (see Examples 3 and 5). However, the result set consists of all the pages on which the matches were found. If both concept label and synonyms are include in the metadata and a tokenizer is used, the result set for simple search for "Niue" is: pages 1,2, 3... in D1, page 3 in* $D2^{+}_{SI}$ *and page 160 in* $D4^{-}_{Niue}$*. Advanced search for "Niue" leads to pages 1,2, 3... in* $D1^{+}_{Niue}$ *and page 3 in* $D2^{+}_{SI}$*. When filtering by concept field, this leads to only pages 1,2, 3... in* $D1^{+}_{Niue}$*, page 3 in* $D2^{+}_{SI}$*. Using an advanced search, e.g., for "Niue", on the full-text in conjunction with the filter by concept selects pages 1, 2, 3... in* $D1^{+}_{Niue}$ *only.*

4.6 Summary of Approaches

The table in Figure 13 provides an overview of the five approaches that were introduced in Section 4.

Approach	semantically enhances		access via		impact on	
	metadata	text index	search	filter	false positives	false negatives
1	x		x	x	--	o
2	x		x	x	-- / +	--
3		x	x		o/+	-
4		x	x		o/++	--
5	x		x	x	-- / +	--

Figure 13: Overview of approaches (Legend: X = applies, empty field = does not apply; –/- = less, +/++ = more, and o = no change

Approaches 1, 2 and 5 add information about semantic concepts to the metadata, while Approaches 3 and 4 add information about semantic concepts to the full-text index. Search is then possible via both the simple interface and the advanced interface for filtering and search in metadata-based enhancements. Figure 13 then summarises our rather complex observations about the influence on false positives and false negatives by simple indications on whether more or less of these results can be expected.

5. TEST COLLECTIONS & DOCUMENTS

This section explores the enhanced indexing approaches for their impact on the size of the resulting indexes, and explores the commonality of the tokens indexed by Solr and the concepts identified by Capisco for a set of example documents.

5.1 Lexical vs semantic-enhanced index

We present the results from experiments testing the potential impact of Approaches 3 and 4 on a digital library's full-text index. We created four test collections from Hathitrust documents, and for each, we created a Solr index and a Capisco index. The Capisco index was then serialised and exported as a concept list for each document, and then imported into the Solr index. Table 2 shows for each collection the number of unique tokens that were identified in Solr and the number of unique concepts that were identified in Capisco. We also list the number of concept labels that refer to terms that already exist in the Solr index (column 'Shared'). The combined size of the index is given both in number of tokens and as a percentage. We also estimated the combined index when including not only the concept labels but also the synonyms. The expanded indexes including synonyms were calculated in their size but not built. On average 5 synonyms per concept are included, but manual extension of the knowledge base may lead to a larger number of synonyms.

The combined index (in comparison to the DL's original Solr index) was between 1% to 10% larger. This is an acceptable increase in size that would not create performance problems for most DL systems. Including the synonyms into the index would also at most double the size of the index, which is manageable for DL systems.

Considering the number of concepts identified for each of these collections, however, highlights the user interface issue for filtering by concept metadata: selecting concepts from a list of more than 2000 concepts that occur within the collection is not feasible. Alternative means of presenting the information may need to be found. Here the disambiguation interface that has been developed for Capisco may provide a suitable starting point.

5.2 Lexical vs semantic analysis

Figure 14 plots the number of lexical tokens (blue) against the number of identified concepts (orange) in 30 documents chosen at random from document collections A–D. The percentage of shared concept/tokens is shown as black circles, using the right-hand axis. For documents D8 and D12, the number of tokens is larger than the maximum of the scale shown in the figure. Furthermore, D8 does not seem to contain any semantic concepts. On closer inspection, we find that Document D8 contains only pages with OCR errors (seemingly random symbols, no text). Thus here Capisco can be used to identify potential problems with OCR-ed text. Documents without any concepts should be automatically flagged for the curator to inspect. Other documents, such as D12, contain surprisingly

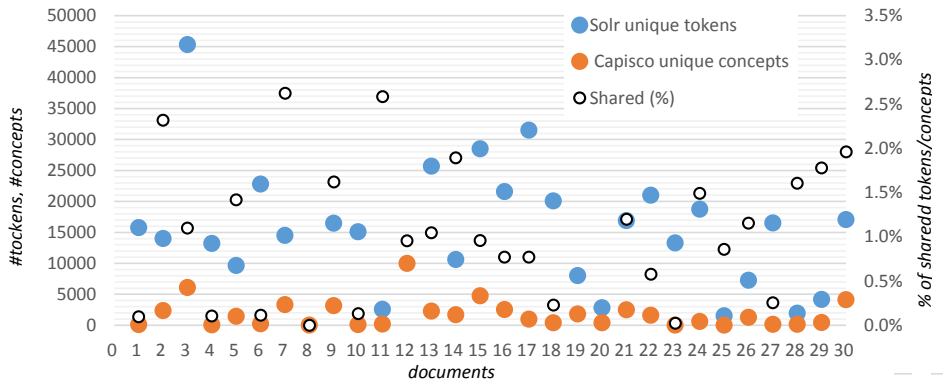

Figure 14: 30 example documents: number of tokens and concepts

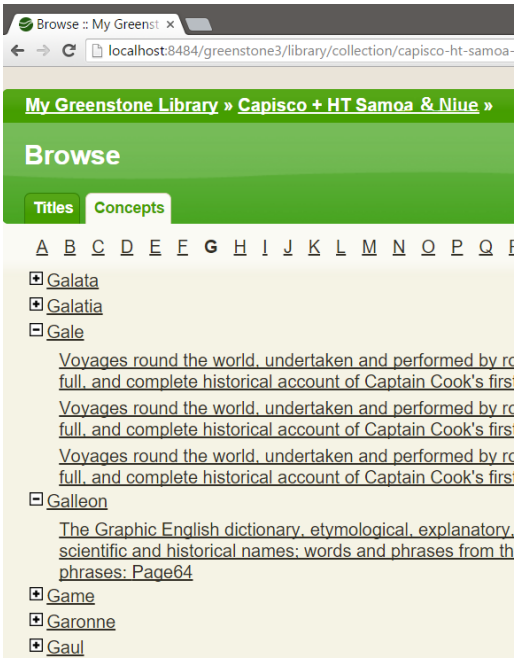

Figure 15: Browsing the assigned concepts: words starting with G

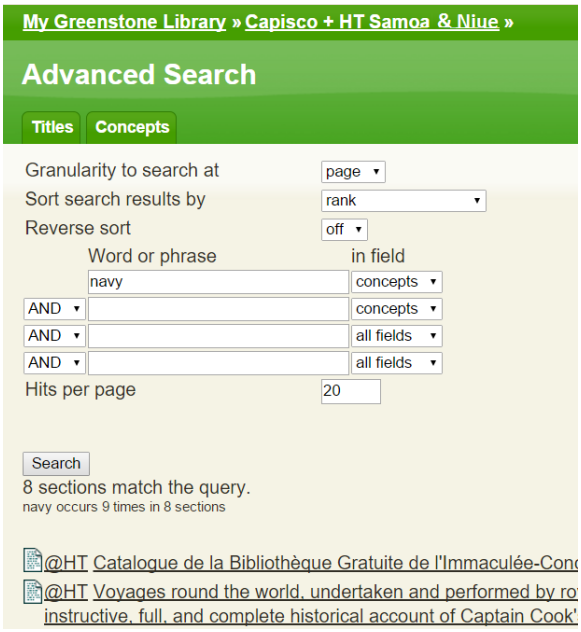

Figure 16: Searching by concept through the advanced search page

few concepts in comparison to the number of tokens. For example, documents D1, D4 and D6 contain 117 concepts vs 15776 tokens, 62 vs 13211, and 210 vs 22804 concepts and tokens, respectively. These documents turned out to be not in English, and the recognised concepts are predominately names of people and locations.[5].

Further research is needed into the significance of particularly low or high commonality between tokens and concepts. These may be indicators of document coherence or writing style (scholarly vs popular writing), or of OCR errors.

6. WORKED EXAMPLE

In this section we present a worked example, based on a Greenstone digital library collection that has incorporated the semantically assigned concepts generated by Capisco when applied to a collection of texts related to Pacific islands Samoa and Niue (collection C). Greenstone allows for parallel indexes to be built. For the example collection, concepts from Capisco were included as a new metadata field at both document level and page level. With the exception of filtering, this structure is rich enough for the user to formulate queries through the advanced search page that covers all the approaches described and analysed in Section 4. In this worked example we concentrate on Approach 5.

Figure 15 shows how these concept metadata fields have been used to introduce a 'browse by concepts' tab to the digital library. The figure shows the example of concepts starting with 'G': the user has clicked on a couple of these (Gale and Galleon) to expand their contents, to view documents containing these concepts.

For Gale, three pages from the book *Voyages round the world* are shown, and for Galleon a page in *The Graphic English dictionary, etymological, explanatory, and pronouncing:...* has been identified.

Browsing helps give the user a feel for what is in the digital library collection. In Figure 16 we have jumped ahead in the timeline of our user's interaction with the DL. In this figure, they have

[5]These collections were created by the HathiTrust using lexical search for research purposes on specific topics regarding the Pacific island nations of Niue and Samoa.

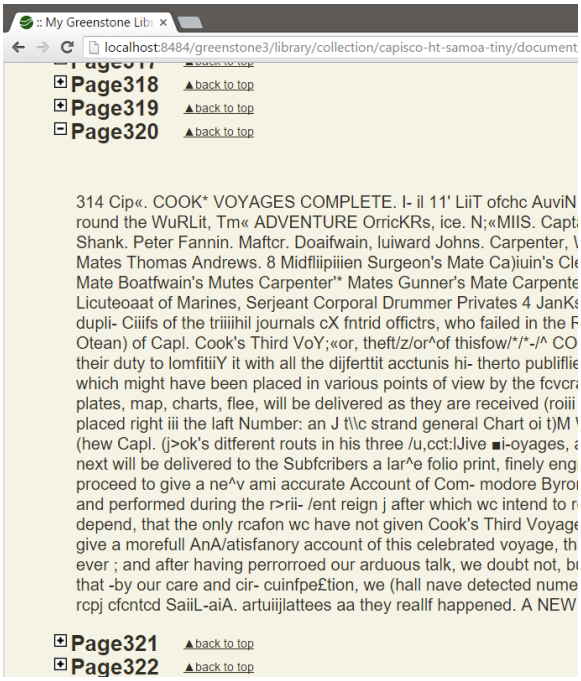

Figure 17: The OCR-ed text of a matching page for concept "navy"

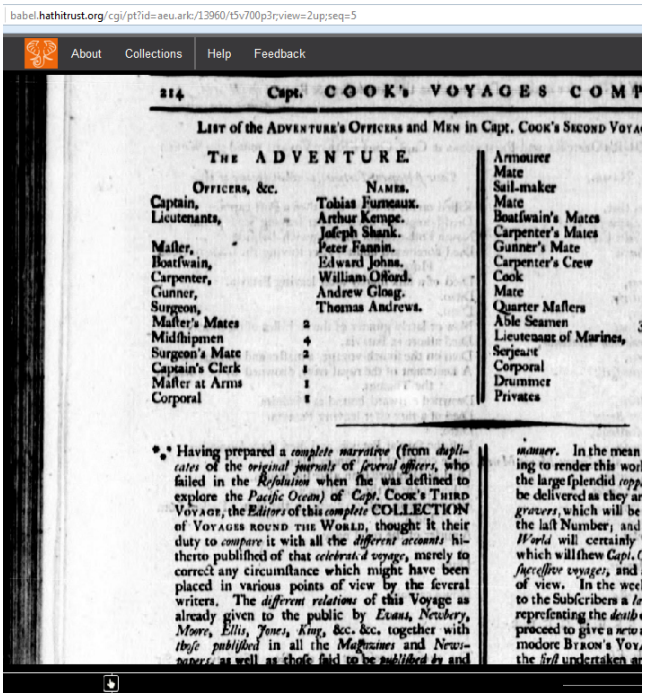

Figure 18: Scanned page corresponding to OCR-ed text in Fig. 17

accessed the advanced search page and configured the interface to search for pages that contain the concept "navy". Shortly we will look at one of the matching documents returned by this query, but first we note the outcome of other searching options available to the user at this point:

- Had the user searched the full-text, then 54 matching pages would have been returned.
- By including the concept labels within the full-text, the number of search results increased to 60.
- Searching by concept label on its own returned 8 pages (the snapshot shown in Figure 16).

From these numbers it can be seen that the concept "navy" has been attributed by Capisco to 2 pages that already have this string literal present, and 6 further pages that do not.

Focusing on the second matching document in the returned result set (the account of Captain Cook's voyages which also appeared in the browsing example), clicking on its link brings up the snapshot shown in Figure 17. In this view, we are shown the OCR-ed text of the document, which is instructive in seeing what text the digital library has actually registered with the page, if not terribly pleasing aesthetically. Using the @HT link on the result page (previous snapshot), the digital library takes the user directly to a scanned version of the document (Figure 18) to view.

Reviewing the information on the page presented, we do indeed see that the page provides details relating to the (British) navy. From the result of the OCR-ed page (the text Capisco has processed), we can see the words and phrases "Marines", "Sail-maker", and "Gunner's Mates" in addition to more generic military terms such as "Captain" and "Lieutenants".

7. DISCUSSION OF APPROACH

Semantic Drift. The accommodation of historical meanings of words, or historical context of words and concepts, can only be achieved if either these are already known to Wikipedia or if these are entered by a scholar who is interested in and aware of the historic meaning of words. Wikipedia does indeed cover some historic concepts, such as the out-of-date usage of "Savage Island" to refer to the Pacific island nation of Niue (as used in our examples). Other aspects may not be covered to the full requirements of a scholar; in these cases the scholars can extend the Concepts-in-Context network as described in [3].

Generalizability of approach. The processes suggested here are directly or with minor adaptations applicable to mainstream digital libraries, such as HathiTrust Digital Library (which was used as our data source) or Greenstone (which was used in our worked example). Any other digital library that uses text-based indexing on full-text and/or indexes on metadata (at page level or document level) would also benefit from the semantic enhancements described here.

Scalability. Test collections have only been explored for Approaches 3 and 4 (see Section 5). Further performance tests are planned for all approaches, including combinations of the approaches. These relatively small test collections are expected to predict the behaviour of *homogeneous* collections adequately. Semantic enhancements of large *heterogeneous* collections are expected to have characteristics similar to those of a combination of smaller homogeneous sub-collections. The performance implications of such combinations will depend on both the homogeneity across the larger collection and the semantic support for each of the sub-collections. These issues will need to be explored further.

Knowledge base seeding. It might appear desirable to use existing ontologies, such as DBpedia, instead of mining Wikipedia. However, these ontologies are too complex for our semantic analysis. They encode rich relationships and hierarchies between concepts. Because of this complexity, they do not cope well with the inherent noise in the documents and would require costly semantic analysis of the sentence structures. The advantage of our CiC network over existing ontologies is its robustness for noise (see [3]). Its simpler semantic analysis is less error-prone and more efficient.

Visualising collection semantics. Enhancing the indexes and metadata of collections with semantic information does not just widen the possibilities for, and quality of, text-based search—it also opens up opportunities to explore the collections. Similar to the existing *ngram* analysis (e.g., [6, 7]), which allows visualisation of changing word use over time, we suggest exploring the semantics over time [2]. While the former allows the linguistic comparison of words, our approach allows the tracking of concepts independent of the actual words used to refer to the concept.

8. CONCLUSION

This paper takes an analytical approach to explore five strategies for low-cost semantic enhancement to DL metadata and indexing. We used a case study of four documents to showcase the differences in result sets for lexical search, semantic search, and each of our five approaches. We also created four test collections for which the performance implications of enriching the full-text index with concept labels was experimentally determined. We additionally estimated the expected growth of the index when using semantic synonyms in addition to concept labels.

We have argued theoretically and shown through examples how using semantic concepts can help identifying OCR errors. Historic documents are particularly susceptible to OCR errors, due to their use of older-style fonts such as *Fractura*, the use of the medial 's' (often misread as f), and general aging and wear of the pages [7]. This aspect has been little explored so far.

Several possible future work directions flow naturally and easily out of having concept information in the full-text index or metadata. Concept information may be useful for relevance feedback on documents. For example, where a user finds a document or page that is of interest to them, they can easily see what semantic concepts were applied. Used this way, the users do not need to "know" the concepts occurring in the digital library or to browse through a list of concept labels. Thus, with no additional effort, query expansion based on semantic concept terms can be supported.

Interface and interaction issues stemming from the joint use of lexical terms and concept labels need to be explored further. For example, since terms in the search may not necessarily match the concept terms, it is not always possible to highlight matching search terms in the document. This is potentially confusing to users.

Further work could also be dedicated to exploring combinations between the approaches introduced in Section 4. For example, a combination of Approaches 2 and 4 promises to offer flexibility and expressiveness to the user.

Even though our work here focused on simply enhancing the lexical search capabilities of traditional digital libraries, the most powerful solution would be a combination between lexical-based search and semantic search as offered in Capisco. Both offer elements that are useful for a scholar; offering merely one of them will always fall short of the full potential.

9. ACKNOWLEDGEMENTS

We thank the recipients of the University of Waikato Summer Research Scholarships, Michael Coleman and Yang Guo, for their work on exploring indexing in Solr and Capisco. The authors thank the Andrew W. Mellon Foundation for their support of this work (grant reference number 41500672). We also thank the staff at the HathiTrust Research Center for their assistance.

10. REFERENCES

[1] H. B. Guppy. *Coral islands and savage myths*. Victoria Institute and Philosophical Society of Great Britain, London, 1889.

[2] A. Hinze, M. Coleman, S. J. Cunningham, and D. Bainbridge. A semantic bookworm: mining literary resources revisited. In *JCDL '16*. ACM, 2016.

[3] A. Hinze, C. Taube-Schock, D. Bainbridge, R. Matamua, and J. S. Downie. Improving access to large-scale digital libraries through semantic-enhanced search and disambiguation. In *JCDL '15*, pages 147–156. ACM, 2015.

[4] A. Huang, D. Milne, E. Frank, and I. H. Witten. Clustering documents using a wikipedia-based concept representation. In *Advances in Knowledge Discovery and Data Mining*, pages 628–636. Springer, 2009.

[5] A. J. Johnes. *Johnes on the causes which have produced dissent from the established church, in the principality of Wales*. Henry Hooper, London, 1870.

[6] P. Leonard. Mining large datasets for the humanities. In *World Library and Information Congress*. International Federation of Library Associations, 2014.

[7] Y. Lin, J.-B. Michel, E. L. Aiden, J. Orwant, W. Brockman, and S. Petrov. Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174. ACL, 2012.

[8] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[9] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.

[10] D. Milne, O. Medelyan, and I. H. Witten. Mining domain-specific thesauri from wikipedia: A case study. In *Proc. IEEE/WIC/ACM international conference on web intelligence*, pages 442–448. IEEE, 2006.

[11] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proc. ACM Conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[12] D. N. Milne, I. H. Witten, and D. M. Nichols. A knowledge-based search engine powered by wikipedia. In *Proc. ACM Conference on information and knowledge management*, pages 445–454. ACM, 2007.

[13] K. Nakayama, T. Hara, and S. Nishio. A thesaurus construction method from large scaleweb dictionaries. In *Advanced Information Networking and Applications, 2007. AINA'07. 21st International Conference on*, pages 932–939. IEEE, 2007.

[14] R. B. O'Brien, editor. *Home Rule, Speeches by John Redmond*. T. F. Unwin, London, 1910.

[15] W. R. Sykes. *Contributions to the Flora of Niue*. Department of Scientific and Industrial Research, Christchurch, 1970.

[16] I. H. Witten, S. J. Boddie, D. Bainbridge, and R. J. McNab. Greenstone: a comprehensive open-source digital library software system. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 113–121. ACM, 2000.