Finding documents and reading them: Semantic metadata extraction, topic browsing and realistic books

© Ian H. Witten, Olena Medelyan and David Milne

Department of Computer Science, University of Waikato, New Zealand
ihw@cs.waikato.ac.nz

Abstract

What would it take to provide a congenial and comfortable environment for finding and reading books in a digital library? To locate information we need algorithms that extract semantic metadata in forms such as keyphrases, with accuracy and consistency comparable to human indexers. To support this we need comprehensive, detailed thesauri, automatically created, that embody contemporary language and usage. To emulate and enjoy the serendipitous adventures found in real libraries and bookstores we need browsing environments that provide readers with multiple clues in parallel: keyphrases, text excerpts, and supplementary knowledge structures—as well as the documents themselves. For readers to cherish and enjoy individual works we need to transcend the bland reading environment provided by the web by recreating the subjective impact and pleasurable experience of interacting with real books. This paper describes research that aims to achieve these goals.

1 Introduction

The facilities for finding and reading information that today's digital libraries provide are dull and primitive. One might think that having the full text of documents at one's electronic fingertips would stimulate radical advances in information location, browsing, and display that far transcend what people do in traditional libraries. But our digital libraries supply little evidence for this. Full-text search—a technology that has no analog in the world of books—is a notable exception, but although indispensable for certain purposes it certainly does not provide a comprehensive foundation for information discovery. Browsing in digital libraries generally rests on manually-derived metadata, not much different from thumbing through an old-fashioned card catalog. When it comes to reading, most online text is perused by scrolling down in a web browser in a manner reminiscent of the papyrus in Ptolemy I's legendary Library of Alexandria founded in 288 BC, four centuries before the development of the codex or book form—a pile of papers held together by a binding. Truly, if the book had been invented after the computer—or after the digital library—it would have been hailed as a stupendous advance.

There is an enormous volume of research on new techniques for finding and reading textual information, but hardly any has gained widespread recognition, let alone adoption. An early goal of the Greenstone Digital Library Software was to serve as a showcase for applied research in information retrieval. Indeed, when we began a decade ago the very idea of large-scale full-text indexing on an ordinary user's workstation was rather revolutionary. We sought widespread adoption of the software so that many could benefit from radical new information-handling techniques that we planned to incorporate. However, the project became mired in dirty practicalities: getting things to work on Windows; making it easy for computer-shy librarians to install; developing the interactive Librarian interface; coping with exploding demands for interoperability on all fronts. Although Greenstone contains traces of novel research—notably the Phind and collage classifiers—we have paid the price of success: demand for extensions and enhancements to the more mundane aspects of the software threatens to swamp our more grandiose ideas.

The present paper describes research directions being pursued by my group in New Zealand, and previews techniques that will eventually find their way into Greenstone. We are working on automatic extraction of keyphrases from documents, and comparing the result with professional indexers. We have developed an algorithm, described in Section 2, for automatically identifying keyphrases: it uses machine learning to determine the most significant phrases describing a document based on their statistical, syntactic, and semantic properties. Section 3 discusses current and proposed work on using lexical chains, which are sequences of semantically related terms that reflect the discourse structure of the text, to improve the keyphrase set by identifying the significant topic areas that a document covers. We believe that automatic techniques may eventually out-perform humans in the consistency and utility of the phrases they extract.

Identifying high-quality keyphrases that describe the documents in a collection requires some knowledge of

Proceedings of the 8th Russian Conference on Digital Libraries RCDL'2006, Suzdal, Russia, 2006

semantics, and in our approach this is supplied by a comprehensive domain-specific thesaurus for the relevant area. Unfortunately good, up-to-date domain-specific thesauri are rare. Section 4 examines the Wikipedia as a large-scale source of contemporary semantic information. We are developing simple ways to tap the enormous potential of this resource by automatically identifying thesaurus relations from it, and find that in a particular domain the result compares favorably with a professionally-produced thesaurus. We are also studying interfaces for interactive query expansion that use a thesaurus to build a bridge between the terminology of the user's query and that used within documents, described in Section 5.

Finally, Section 6 reviews work on ways of allowing users to view and interact with realistic three-dimensional book-style visualizations of any text-based document in a digital library collection. Physical book models offer readers something beyond traditional computer-based paging or scrolling systems, and can be enhanced with metadata to further enrich the browsing experience.

2 Keyphrase indexing

Keyphrases, a brief but precise summary of documents, are widely used for organizing library holdings and providing thematic access. Assigning them manually is expensive and time-consuming, so automatic techniques are in great demand. Existing approaches to keyphrase indexing include *extracting* significant phrases from documents on the basis of properties such as frequency and length [1, 2, 3], and *assigning* documents to keyphrases from a controlled vocabulary with the help of a large set of training documents [4]. While extracted phrases are often ill-formed or inappropriate, keyphrase assignment requires extensive manually-indexed training data. Our system KEA++, based on a predecessor KEA [5], takes an intermediate approach that circumvents these limitations. It maps document phrases onto terms in a controlled vocabulary—a thesaurus—and learns significant features from a small set of manually-indexed documents.

2.1 How KEA++ works

Each document in the collection is segmented into individual tokens on the basis of white space and punctuation. All word n-grams that do not cross phrase boundaries are extracted, normalized and matched against the controlled vocabulary. Terms with equivalent meaning are recognized using the thesaurus's synonymy relations, and non-preferred terms are replaced by the corresponding preferred descriptor. The resulting candidate set consists of grammatical terms that relate to the document's content.

The next step is to identify the most important of these candidates. A set of training documents with keyphrases assigned by professional indexers is used to build a model. For each document, candidate terms are identified and their feature values calculated. Four features turned out to be useful in our experiments: the TF×IDF score, the position of the first occurrence of the phrase, its length in words, and the node degree—that is, the number of thesaurus links that connect the term to other candidate phrases. If a document describes a particular topic area, it covers most related thesaurus terms, so phrases with high node degree are more likely to be significant.

Table 1. Performance of KEA, KEA+ and KEA++

	P	R	F
KEA	13.3	12.4	12.0
KEA+	20.5	19.7	18.7
KEA++	**28.3**	**26.1**	**25.2**

Each candidate phrase in a training document is marked according to whether or not it was manually assigned as an index term. This binary feature is the "class" used for machine learning. The learning scheme generates a model that predicts the class using the values of the other features. We use the Naïve Bayes technique because it is simple and yields good results. It learns two sets of numeric weights, one applying to positive instances ("is an index term") and the other to negative ones ("not an index term").

To select index terms from a new document, candidate terms and their feature values are determined, then the model built during training is applied. The model determines the overall probability that each candidate is an index term. The top-ranked candidates are selected as the final set of index terms.

2.2 Evaluation and examples

The training and evaluation material comprises 200 full-text documents that were downloaded from the UN Food and Agriculture Organization's document repository (*www.fao.org*). Agrovoc is a domain specific thesaurus that FAO uses for indexing [6]. It contains 16,600 descriptors and 10,600 non-descriptors linked to synonymous descriptors, and defines hierarchical and associational semantic relations between descriptors. Each document had been manually indexed with an average of 5.4 Agrovoc terms.

We compared KEA++ with KEA on this document set by estimating the number of matching ("correct") keyphrases. We expressed this as a proportion of all extracted phrases (Precision *P*) and of all manually assigned phrases (Recall *R*) for each document separately; the F-measure is a balanced combination of the two. Table 1 presents average values over all documents using 10-fold cross-validation. The main finding is that KEA++ roundly outperforms its predecessor, achieving more than double the level of recall, precision, and F-measure.

Part of the improvement is due to the use of a controlled vocabulary: KEA extracts 14 times more candidates and therefore has more difficulty filtering them. To quantify this, results are shown for an intermediate algorithm KEA+ that performs controlled indexing but just uses the original two features. The new features—length and node degree—together gain a further 6 to 8 percentage points.

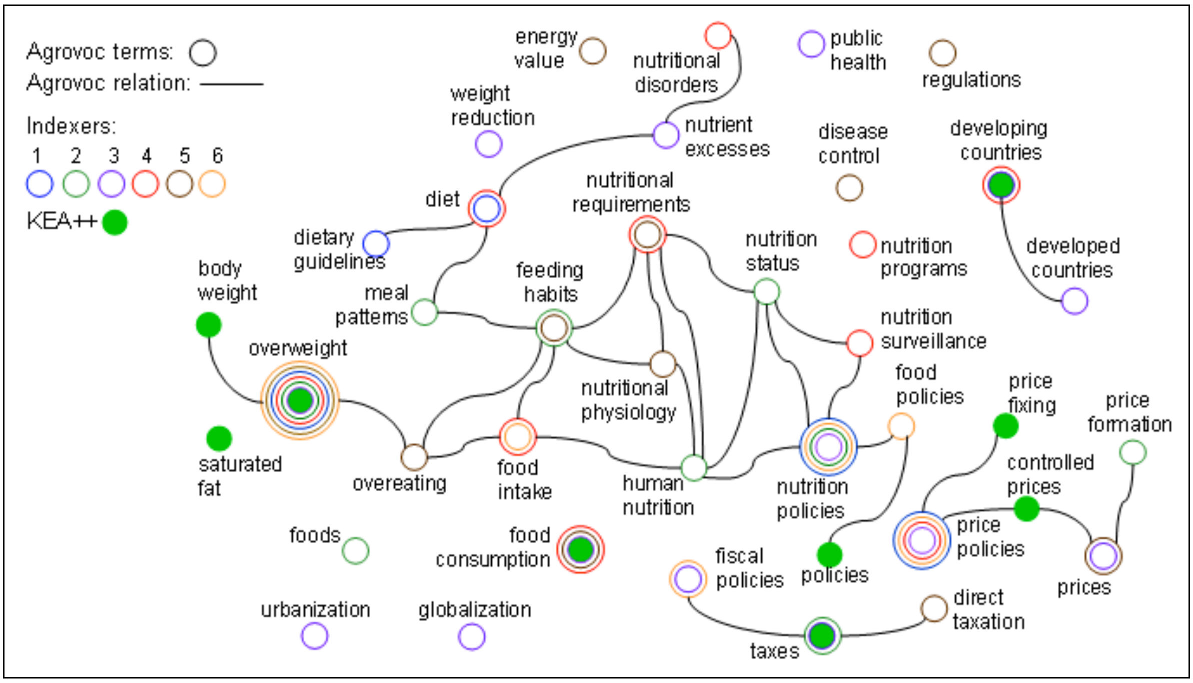


Figure 1. Keyphrases assigned by six professional indexers and by KEA++

Indexing is a subjective task: even professionals assign different terms to a given document. Keyphrases assigned by just one indexer are not the only "correct" ones. We propose that the gold standard in indexing is the degree of *inter-indexer consistency* reached by several professional indexers working independently; this quantifies their agreement on index terms. Our goal is to develop an automatic indexing method that is at least as consistent with a group of human indexers as they are with each other.

We obtained a second collection containing 10 documents indexed by 6 professional FAO indexers, with an average of 9.6 terms per document and indexer. We computed their consistency using Rolling's measure [7], and applied the same measure to keyphrases assigned by KEA and KEA++, after being trained on the 200 documents in the main collection. The humans achieved an average consistency of 38%. KEA++ achieved 27%, an impressive result—particularly when compared with KEA's 7%. In these terms, the new algorithm is not far off human performance.

Figures 1 shows keyphrase sets assigned to a document entitled *The growing global obesity problem* by the 6 FAO professionals (circles) and KEA++ (solid centers); the lines joining nodes indicate thesaurus relations. Human indexers disagreed on most of the keyphrases (single circle). They all agreed on only one, *overweight*, which KEA++ also selected. KEA++'s other choices are either the same or similar to those chosen by the professionals.[1]

3 Improvements through lexical chaining

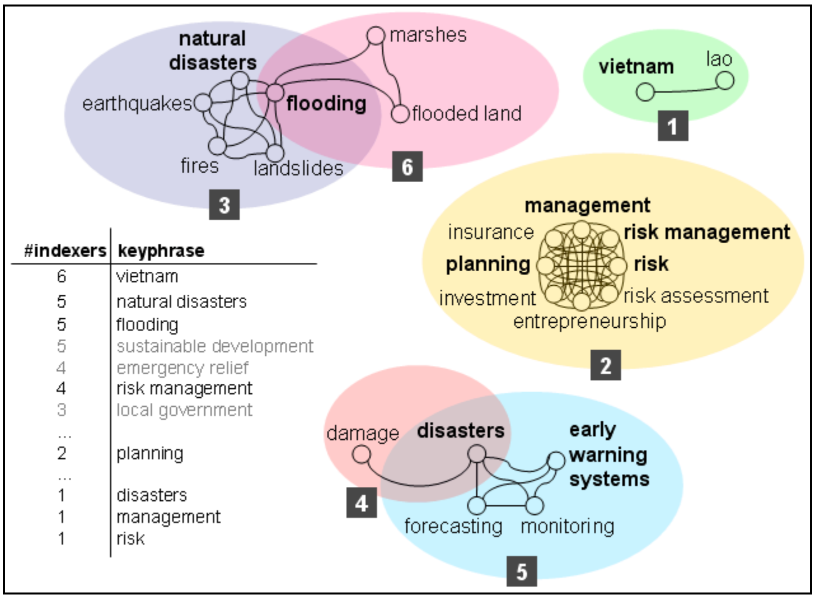
KEA++ treats each candidate phrase independently. It often extracts several keyphrases describing the same topic (e.g. *price fixing* and *controlled prices* in Figure 1), and omits phrases corresponding to other important topics (e.g. *nutrition policies*). To address this deficiency we are using lexical chains to group semantically related phrases into coherent sequences.

Lexical chains express the cohesive structure of a document, and have been applied to such tasks as automatic text summarization [8, 9], topic detection [10], and document structuring [11]. They are constructed using a step-by-step analysis of each word in a document and its semantic relation with other, already-processed, words. The overall strength of each chain is computed by a weighting function that takes into account the number of its members, their total occurrence frequency and the kind of semantic relations between them. A new word is included in the existing chain to which it contributes the most weight. If it is unrelated to all words in existing chains, it begins a new one. Semantic relations between words are obtained from an electronic thesaurus such as WordNet [12] or Roget.

The information that these chains reveal about a document is particularly useful for automatic indexing. Each one, computed over the full text, represents a single topic; those with the highest scores correspond to the major topics a document discusses. Strong members of strong chains are potential keyphrases.

We plan to exploit these characteristics as follows. First we will design a weighting function that returns similar score distributions for document of any length. Then we will analyze all chains whose score exceeds a predefined threshold. Depending on their strength, one or more members will be included in the final keyphrase set. This technique will take the document's coverage and specificity into account when assigning keyphrases. The weight of the chain containing a

[1] More examples are at *www.nzdl.org/Kea/Kea-4.0.html*



#indexers	keyphrase
6	vietnam
5	natural disasters
5	flooding
5	sustainable development
4	emergency relief
4	risk management
3	local government
…	
2	planning
…	
1	disasters
1	management
1	risk

Figure 2. Six lexical chains, and the keyphrases assigned by human indexers

candidate phrase is likely to be a valuable feature, for it quantifies the importance of the corresponding topic.

We have performed some initial experiments with lexical chains by implementing an algorithm that computes them from agricultural documents using the Agrovoc thesaurus. A word is included in a chain only if it is semantically related to every other chain member. This produces stronger, more coherent chains. Since we are dealing with domain-specific documents, polysemy is rare, so, like [9], we do not disambiguate word senses; instead we allow each word to appear in more than one chain.

We found that FAO documents contain numerous lexical chains, but most are singletons and can be ignored. For each document there are only a few long chains with frequently occurring members. For example, 10 chains with three or more members were extracted from a report entitled *Overview of Techniques for Reducing Bird Predation at Aquaculture Facilities*:

{birds, predatory birds, vertebrates}	112
{uses, management, efficiency}	65
{ponds, beaches, valleys, highlands, plains}	34
{aquaculture, fisheries, aquaculture techniques}	31
{canada, alberta, ontario}	24
{electronics, mechanics, physics}	11
{fires, winds, storms}	9
{attractants, control methods, repellents}	9
{topping, rolling, ridging}	4
{claws, scales, horns}	3

The numbers represent the total frequency of each chain's members. The document contained 245 further chains with ≤2 members.

To increase the chains' coverage and quality we plan to introduce additional resources that cover more document phrases than Agrovoc, which is a fairly small thesaurus. However, even in its current state the chaining algorithm yields useful results. We compared the six strongest chains extracted from a document entitled *Role of Local Institutions in Reducing Vulnerability to Natural Disasters: Vietnam* with keyphrases assigned by 6 professional indexers. Figure 2 shows that they cover most of the phrases selected by most indexers. Each chain represents a topic in the document that corresponds to one chosen by human indexers. We believe that the use of lexical chains will greatly enhance the quality of automatic indexing.

4 Wikipedia vs. domain-specific thesauri

These techniques, and the browsing environment described next, require a high-quality domain-specific thesaurus. How can you obtain a thesaurus to support a library of documents relevant to a given domain? Manual construction is prohibitively expensive; automatic generation is woefully inaccurate. General thesauri do not incorporate the specialist terminology that pervades our professions, nor can they keep pace with the deluge of new topics and concepts that arrive daily. Yet a contemporary resource that incorporates expertise in all fields of human endeavor already exists: the collaborative encyclopedia Wikipedia.

Thesauri serve as controlled vocabularies that bridge the variety of idiolects and terminology present in a document corpus. Each concept is named by a "preferred term" to which alternative expressions are linked via the synonymy relation. Likewise Wikipedia ensures a single article for each concept by using "redirects" to link equivalent terms to a preferred one, namely the article's title. It copes with capitalization and spelling variations, abbreviations, synonyms, colloquialisms, and scientific terms. The top left of Figure 3a shows four redirects for *library*: the plural *libraries*, the common misspelling *libary*, the technical term *bibliotheca*, and a common variant *reading room*.

Scope notes delimiting the meaning of each term help users disambiguate ones that relate to multiple concepts. Wikipedia provides disambiguation pages that present various possible meanings from which users

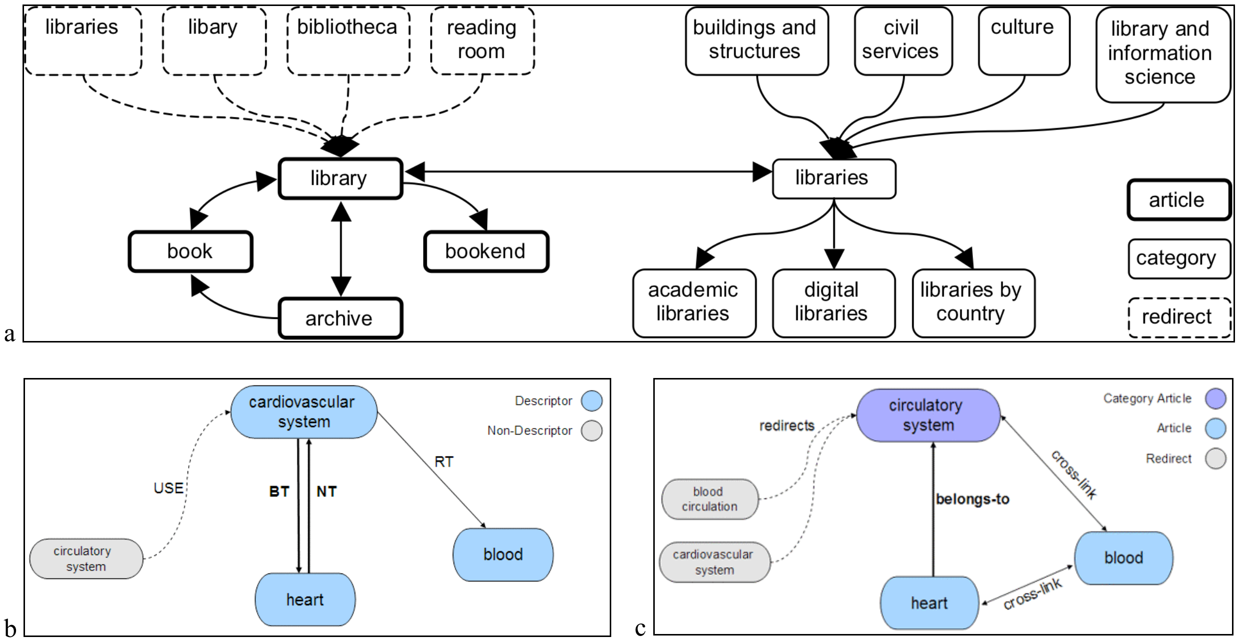


Figure 3 (a) Example structure from Wikipedia
(b) Agrovoc: Descriptors, non-descriptors, semantic relations
(c) Wikipedia: Categories, articles, redirects, and links

select the intended article. For example, the term *library* yields these options:

- *Library*, a collection of books
- *Library (computer science)*, a collection of subprograms used to develop software
- *Library (electronics)*, a collection of cells, macros or functional units that perform common operations
- *Library (biology)*, a collection of stable molecules that represents some aspect of an organism.

The articles themselves serve as detailed scope notes—they fully describe the intended meaning of the term.

The hierarchical organization of terms in a thesaurus is reflected in Wikipedia's categorization structure. Authors are encouraged to assign categories to their articles, and the categories themselves can be assigned to other, more general, categories. The right-hand side of Figure 3a shows a structure in Wikipedia that exemplifies these principles. The article *library* has a corresponding category *libraries*, which contains several more specific subcategories and articles, such as *academic libraries* and *digital libraries*. Other categories, such as *libraries by country*, have no corresponding articles and serve only to organize the content. Both articles and categories can belong to more than one category. *Libraries* belongs to four: *buildings and structures*, *civil services*, *culture*, and *library and information science*. Wikipedia's category structure does not form a simple tree-structured taxonomy but is a graph in which multiple organization schemes coexist.

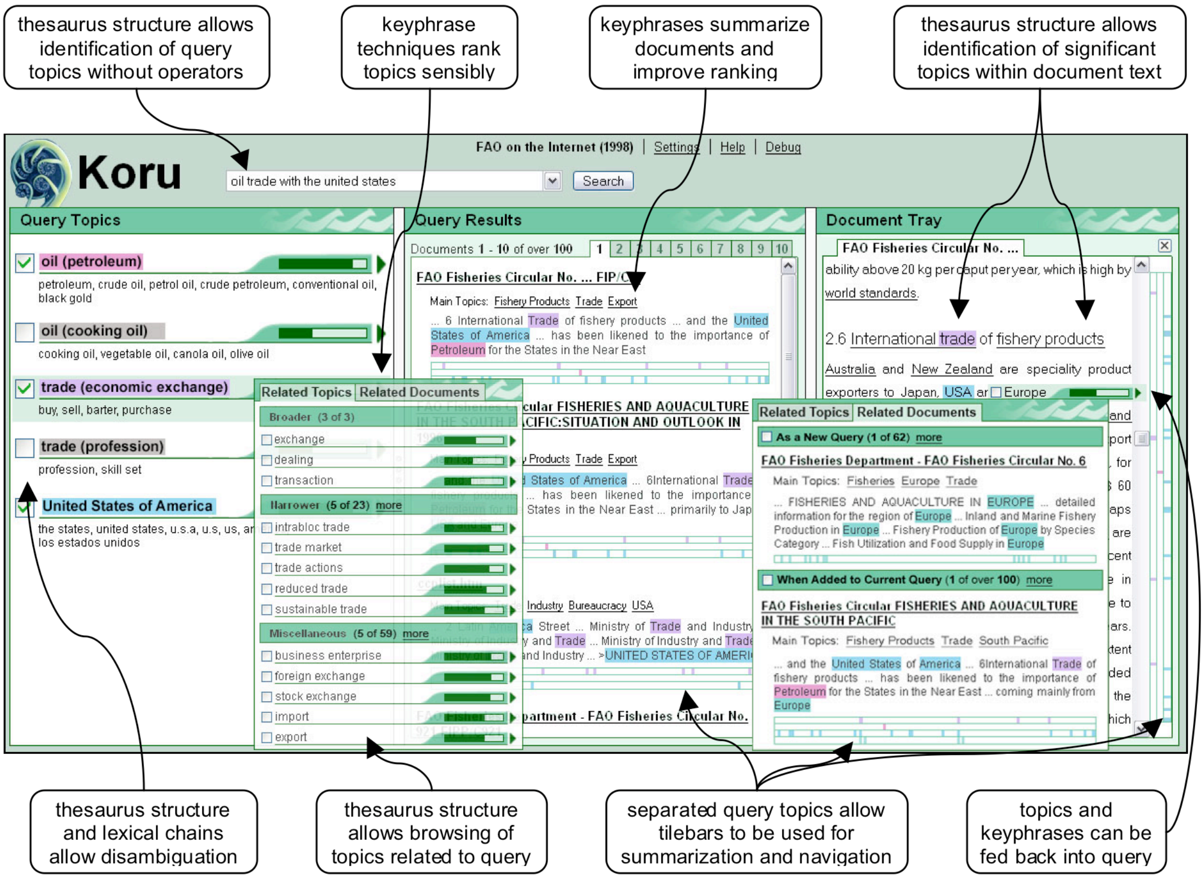
Hyperlinks in Wikipedia express relatedness between articles. For example, the lower left of Figure 3a shows hyperlinks between the article *library* and those for *book*, *archive*, and *bookend*; some of these articles link back. Articles are peppered with such connections, which can be explored to mine the associative relations that thesauri contain.

There are two problems: links often occur between articles that are only tenuously related, and there is no explicit typing of links. The first issue can be ameliorated by considering only mutual hyperlinks between articles—we call them "cross-links." This discards the putative associative relation between *library* and *bookend* in Figure 1. As for the second, we must seek clues as to whether the relation is hierarchical or associative. If it already occurs within the category structure, it must be hierarchical. Statistical and lexical analysis could also be used: for example, the *library* article has many more links and is therefore broader than *archive*.

Figures 3b and 3c show parallel structures for the same concepts in Agrovoc and Wikipedia. Though Agrovoc chooses *cardiovascular system* and Wikipedia *circulatory system* as the descriptor, both cite the other term as a synonym; Wikipedia also cites the more informal term *blood circulation*. The broader/narrower (BT/NT) relation with *heart* in Agrovoc has a parallel relation in Wikipedia's category structure; cross-links correspond roughly to Agrovoc's related terms (RT).

The English version of Wikipedia contains a million articles (descriptors) and a further million redirects (non-descriptors); Agrovoc has 16,600 and 10,600 respectively. Direct comparison of terminology reveals that Wikipedia covers approximately 50% of Agrovoc. The vast majority of terms found in the former but not the latter lie outside the domain of agriculture. Cursory examination of Agrovoc terms not covered by Wikipedia indicates that they are generally scientific terms or highly specific multi-word phrases such as *margossa*, *bursaphelenchus* and *flow cytometry cells*.

Wikipedia's redirects cover 75% of Agrovoc's synonymy (descriptor–non-descriptor) relations; a further 20% of related term pairs that Agrovoc deems



thesaurus structure allows identification of query topics without operators

keyphrase techniques rank topics sensibly

keyphrases summarize documents and improve ranking

thesaurus structure allows identification of significant topics within document text

thesaurus structure and lexical chains allow disambiguation

thesaurus structure allows browsing of topics related to query

separated query topics allow tilebars to be used for summarization and navigation

topics and keyphrases can be fed back into query

Koru
FAO on the Internet (1998) | Settings | Help | Debug
oil trade with the united states
Search
Query Topics
oil (petroleum)
petroleum, crude oil, petrol oil, conventional oil, black gold
oil (cooking oil)
cooking oil, vegetable oil, canola oil, olive oil
trade (economic exchange)
buy, sell, barter, purchase
trade (profession)
profession, skill set
United States of America
the states, united states, u.s.a, u.s, us, am, los estados unidos
Query Results
Documents 1 - 10 of over 100
FAO Fisheries Circular No. ... FIP/C...
Main Topics: Fishery Products Trade Export
Related Topics Related Documents
Broader (3 of 3)
exchange
dealing
transaction
Narrower (5 of 23) more
intrabloc trade
trade market
trade actions
reduced trade
sustainable trade
Miscellaneous (5 of 59) more
business enterprise
foreign exchange
stock exchange
import
export
Document Tray
FAO Fisheries Circular No. ...
2.6 International trade of fishery products
As a New Query (1 of 62) more
FAO Fisheries Department - FAO Fisheries Circular No. 6
Main Topics: Fisheries Europe Trade
When Added to Current Query (1 of over 100) more
FAO Fisheries Circular FISHERIES AND AQUACULTURE IN THE SOUTH PACIFIC
Main Topics: Fishery Products Trade South Pacific

Figure 4. Browsing topics and documents related to *oil trade with the united states*

equivalent are encoded in Wikipedia through other links. Examples indicate that Wikipedia separates such pairs into distinct articles rather than treating them as synonyms, e.g. *aluminum foil* → *shrink film* and *spanish west africa* → *rio de oro*. Agrovoc judges these concepts to be "near enough" not to require separate entries, whereas Wikipedia is more discriminating.

Wikipedia covers 69% of Agrovoc's hierarchical (BT/NT) relations. Only 25% appear in the category structure; the remaining 44% are found in redirects and hyperlinks between articles. However, preliminary results indicate that coverage doubles when the transitive nature of links is taken into account—for example, the relation *oceania* → *american samoa* is implied by the chain *oceania* → *oceanian countries* → *american samoa*. It is also possible to mine relations found elsewhere, but this would require additional analysis to identify the direction of the relation. For example, a hyperlink between two articles does not say which is broader and which is narrower. This information may be encoded textually (e.g. *South Africa* is a lexical expansion of *Africa*) or statistically (e.g. *forestry* has many more links than *logging*).

A full 84% of the relations in Wikipedia's category structure are absent from Agrovoc's hierarchy. Many are implied by transitivity; others are irrelevant to Agrovoc's domain. The remaining relations form a useful increase in connectivity.

Wikipedia covers 56% of Agrovoc's associative (RT) relations. Mutual links between articles account for 22%; the remaining 34% are found within one-way links or the category structure. Also, only 5% of mutual article links correspond to Agrovoc RT relations. Many describe relations that Agrovoc leaves implicit, e.g. all siblings are implicitly RTs. Other mismatches may be caused by inadequate sense disambiguation. As with hierarchical relations, extracting thesaurus-style RTs is a complex procedure that requires sense disambiguation and examination of other link locations in Wikipedia.

We believe that useful domain- and corpus-specific thesauri can be derived from Wikipedia by intersecting its phrases with those found in the documents. Comparing terms and semantic relations to a manually created domain-specific thesaurus demonstrates excellent coverage of domain terminology, and of synonymy relations. Wikipedia is a good source of hierarchical and associative relations, with scope for improvement in coverage and accuracy.

5 Browsing phrases, keyphrases, topics

Koru is the Māori word for the newborn, unfurling fern frond; a delicate spiral of expanding fractal shapes. For indigenous New Zealanders it symbolizes growth; rebirth; evolution. Likewise, the Koru topic browsing system, building on the work described above, provides an environment in which users can progressively work towards the information they seek.[2] It blurs the lines between browsing and searching by allowing seekers to incrementally evolve their queries and encounter new topics and terminology in a rich variety of ways.

Koru, illustrated in Figure 4, is based on the AJAX framework [14], which provides a highly reactive interface couched in nothing more than the standard elements of a webpage. The uppermost area is a classic search box in which the user has entered the query *oil trade with the united states*. Below is a triptych of panels; *query topics*, *query results*, and a *document tray*.

What the Figure does not convey is that to avoid clutter at most two of the three are visible at any given time. There are three possible configurations, which relate to three stages of expected user behavior:

1. *Building an appropriate query.* This involves adding and removing phrases until the query and resulting list of documents satisfies the user's information need. At this stage query topics and query results are visible.
2. *Browsing the document list.* Once a suitable list of documents is returned, the user must determine the most relevant ones and judge whether they warrant further reading. At this point the panels in Figure 4 slide across so that only the query results and the document tray are visible.
3. *In-depth reading of the document.* Having located a worthy document, the user then devotes time to actually reading the relevant sections. Here only the document tray is needed; anything else would be a distraction.

5.1 The query topics panel

The query topics panel provides users with a summary of their query and a base from which to evolve it. It lists each significant topic—i.e., keyphrase—extracted from the query, and assigns to each a color that is used consistently throughout the interface. These topics are obtained without requiring any special query syntax using a corpus specific thesaurus extracted with the techniques of Section 4. The thesaurus covers both document terminology and contemporary usage, and is thus likely to cover potential query terms. The thesaurus also identifies synonymous terms for each topic (listed below the topic), which are automatically incorporated into the query to improve recall.

Query terms are often ambiguous and relate to multiple entries in the thesaurus, as was the case for *oil* and *trade.* Each possible sense is ranked according to the likelihood that it is a key topic for the document collection (displayed as a horizontal bar to the right of each topic), using the keyphrase indexing techniques described in Section 2. Only the top-ranked phrase is selected automatically; this can be overridden using checkboxes to the left of each topic. This improves the query's precision by discarding documents that do not use the term in the intended sense. Document level sense disambiguation is achieved using lexical chains (Section 3) as described in [13]. This is critical when automatically including similar terms, which are often ambiguous and would otherwise reduce precision.

Each topic can be used as an entry point into the thesaurus structure. In Figure 4, the arrow next to *trade (economic exchange)* has been clicked to reveal a menu of related topics. These are separated into three groups: broader topics such as *exchange* and *transaction* can be used to make the query more general; narrower ones such as *intrabloc trade* and *trade market* make it more specific; and miscellaneous topics such as *business enterprise* and *foreign exchange* allow the user to switch to related domains. Each topic in these lists is presented with exactly the same controls as in the query topic panel: it can easily be evaluated, incorporated into the query, or used to explore related topics.

5.2 The query results panel

The query results panel presents the results of the query in the form of a series of document surrogates. The first elements of each surrogate are the document title and a set of keyphrases (described in Section 2). Although these succinctly express the nature and content of the document, it is often sections that are relevant rather than the document as a whole. This information is conveyed by snippets that reflect the document's relationship to the query. Within both titles and snippets, query terms are highlighted for ease of identification.

An overview of how query topics are distributed is presented graphically using tilebars [15]. These represent the entire content of the document as a horizontal bar from left (beginning of document) to right (end). Different bars provide a separate region for each query topic. Points are distributed along these bars to represent the locations of phrases from the equivalent topic. Parts of the document where topics are clustered and different topics co-occur stand out visually.

5.3 The document tray

The purpose of the document tray is to facilitate efficient reading by aiding identification of and navigation between relevant sections of documents. These sections are identified using the same information that made the document itself relevant: the query terms used to locate it. Term occurrences are highlighted according to the colors defined in the query topics panel, so they can be identified by a quick visual scan. Interesting patterns of highlights are likely to indicate sections and paragraphs that should be read.

These highlights can easily be missed, however, because most documents are too large to be viewed without scrolling. To counter this, tilebars are supplied to provide an overview of how terms are distributed throughout the document. These tilebars are oriented vertically, with a direct mapping to the standard scrollbar. If the scrollbar slider is moved alongside a cluster of points in the tilebar, the highlights that these points represent are made visible. It is also possible to jump directly to a particular highlight by clicking the appropriate point in the tilebar.

[2] More information at *greenstone.org/greenstone3/koru*

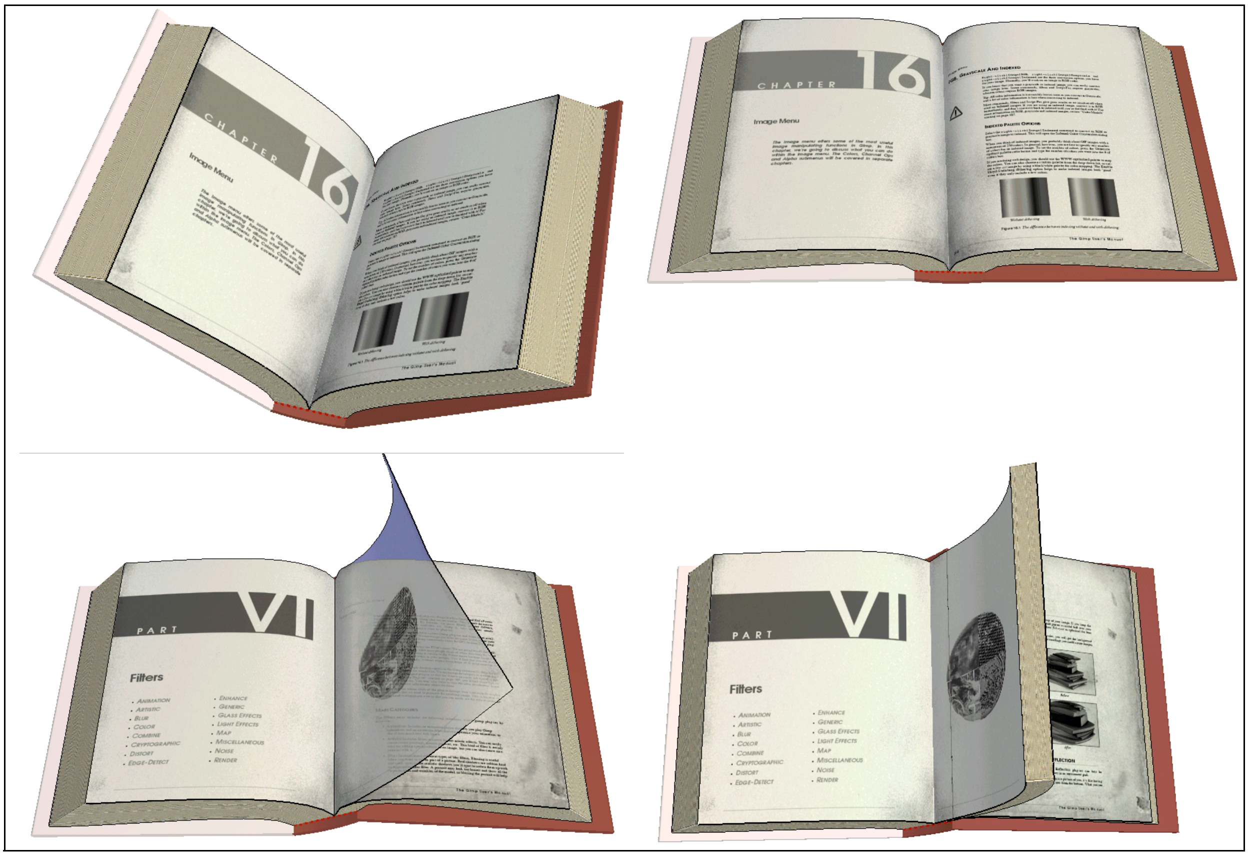


Figure 5. A realistic book

Documents are rich sources of new query topics, which are highlighted as links interpolated throughout the document. These may be strongly related to the existing query (such as *international trade*), representative of document content (such as *fishery products*), or significant topics throughout the collection (such as the countries listed). They can be moused over to obtain the same controls provided for other potential topics: they can be evaluated, incorporated into the query, or explored for related topics. In Figure 4 the user has chosen to explore documents related to *europe*. They can explore the topic as an independent query, locate documents that describe it best, or preview how its inclusion would affect the current query.

6 Open The Book: A 3D book visualizer

Notwithstanding the convenient browsing facilities of Koru's document tray, for many readers handling a physical book is an enjoyably exquisite part of the information seeking process. Physical characteristics of a book—its size, heft, the patina of use on its pages and so on—communicate ambient qualities of the document it represents. In contrast, the experience of accessing and exploring digital library documents is dull. The emphasis is utilitarian; technophile rather than bibliophile. We are developing a scaleable, systematic approach that allows users to view and interact with realistic visualizations of any textual-based document in a Greenstone collection. Our work leads us to believe that far from being a whimsical gimmick, physical book models can usefully complement conventional document viewers and increase the perceived value of a digital library system.

Figure 5 shows a simulated 3D book in various poses. The page contents are texture-mapped onto the "paper" using standard computer graphics techniques, so the same simulated model can represent any particular book. In our prototype a wide range of parameters are stored in a configuration file. Viewing parameters range from the size of the window that opens to the amount of ambient light present in the world and the user's initial viewpoint. Book parameters range from the number and thickness of pages to the material properties (diffuse, reflective, etc.) of the paper and cover.

People often heft objects to literally "get a feel" for them. In this implementation the mouse is used to control the book's orientation. Although there is no sense of weight or touch, one nevertheless gains the distinct impression of handling a physical object. The cover (and pages) of the book can be colored and/or textured to visually represent properties such as age.

You cannot riffle through the pages of this book with your fingers. But you can click on the edge of the book, grab it at a certain position, and open it using mouse or touch-screen. In the first image the user has, with the left mouse button, grabbed a page edge about three-quarters of the way through and started to open the book at that point by moving the mouse to the left. The opening follows the mouse as it sweeps leftwards. Alternatively, the user can click on the front cover and sweep left to open the book at the first page. Repeating the process on the newly exposed right-hand page performs the next page turn; clicking and dragging the facing page returns the user to the previous page.

With the book slightly more than half open, the user lets go and the left part continues to fall, eventually coming to rest as shown in the second image. The user could equally well have continued the opening sweep, bringing the book to its final position with their grip still on it, or reversed the direction of the sweep partway through and returned the book to a closed position. If the book was less than halfway open when it was released, it falls closed. The model behaves the same way regardless of how many pages are being turned.

The behavior of the spine during a page turn is central to the credibility of the visualization, and we have taken pains to model this detail accurately. Notice how the pages bend into the spine, and how the angle of the spine relative to the cover changes as the book opens further. Although you cannot really see this in these static figures, the spine even bows slightly during the turning process to adjust to the pressure that the two covers exert on it, just as a physical book does.

The user continues to examine the book. By the time the third image is reached, they have settled into viewing a sequence of consecutive pages, involving single page turns. The sweeping and releasing actions are the same as for opening the book: you initiate a page-turn by clicking on the exposed page. Interaction is mode-less: users can switch between the various forms of page access at will. Incidentally, as the figure illustrates, the pages are slightly transparent, but they do not have to be. In the fourth image the user has once again used the page edges to shift to a later section of the book. Rather than clicking the next page, they have clicked further through the book. Compare the curvature of the turning pages with the rigidity of the hardback cover in the first image. The curvature varies as the turn progresses.

As well as imitating nature by modeling physical aspects of books, the book visualizer also incorporates artificial features that are based upon visual metadata. For example, markers can appear down the right-hand side of the book, which correspond to the start of each chapter. Clicking in the page-edge region now snaps to the closest marked point and opens the book there. In our model, snapping to anchor points and placing marks down the edge of the book can be independently toggled on or off. When snapping is on but the markers are off, whenever the book is opened it appears to "coincidentally" open to the beginning of a chapter, without any visual distraction.

Visual effects of aging can be linked to the book's usage log. Pages that have been accessed more than others appear grubbier around the edges.

You can begin flipping through a book by making a sweeping gesture with the right mouse button. This begins an animation in which individual pages, or groups of pages, turn one by one until the end is reached. The larger the gesture, the more pages are turned per flip. An alternative to fixed page flipping is to use anchor points, resulting, for this book, in a traversal of the book chapter by chapter. Uninterrupted, the activity continues until the end of the book has been reached, whereupon the book closes.

It is easy to imagine further enhancements. Section headings could pop out of the side of the book, either to form a complete table of contents that is keyed to physical locations within the book, or as rollover text when moused. Many user interface aids operate on a document surrogate—e.g. tilebars display the result of searches by coloring areas that contain hits to the various search terms—and these adapt very naturally to a physical book model, where the edges of pages are colored to indicate clusters of terms visually. A conventional index, or index of automatically-extracted keyphrases, could be keyed to page locations in the book. Realistic documents in a digital library can do far more than simply mimic their physical counterparts.

7 Conclusions

We aim to provide the elements of a congenial and comfortable environment for finding and reading books in a digital library. Patrons will be able to browse around the library using information distilled from the full-text contents of the documents it contains. To offer help on the contents that is meaningful rather than superficial, it is necessary to analyze the semantics of the documents. We need a comprehensive domain-specific thesaurus to supply semantic information. If one does not exist, we plan to leverage the content of WordNet and the Wikipedia instead.

With its interwoven tapestry of articles in many languages, Wikipedia is a huge mine of valuable information about words and concepts; its exploitation is just beginning. While it has attracted criticism [16], these concerns are for the most part irrelevant to our purposes and are far outweighed by many advantages that traditional resources cannot possibly offer.

We are developing a new kind of browsing structure by integrating phrases distilled from the document text with other knowledge sources such as thesauri. This seamlessly combines elements of full-text search and query expansion with more traditional browsing around a topic hierarchy. Library patrons can continue their explorations right into the documents; textual snippets in the search results and tilebars within the documents themselves maintain the context of their browsing and relate it to the topics they are examining. Progress in AJAX technology makes it possible to offer end-users a fast, responsive, and attractive environment for browsing the contents of a digital library.

Finally, we believe that people have valuable and enjoyable interactions with books as objects, without necessarily reading them from cover to cover. And even when they do read a book right through from beginning to end, they invariably take a good look at the outside—and inside—first. The scrolled or paged documents we read from our web browsers are but pale electronic shadows of real books. We think users should be offered something that is far closer to the real thing—only better.

References

[1] Barker, K., Cornacchia N. (2000) “Using noun phrase heads to extract document keyphrases.” *Proc Canadian Conference on Artificial Intelligence*, 40-52.

[2] Hulth, A. (2004) *Combining machine learning and natural language processing for automatic keyword extraction*. Ph. D. thesis, Stockholm University.

[3] Turney, P. (1999) “Learning to extract keyphrases from text.” Technical report, National Research Council of Canada.

[4] Dumais, S.T., Platt, J., Heckerman, D., Sahami, M. (1998) “Inductive learning algorithms and representations for text categorization.” *Proc ACM-CIKM*, 148-155.

[5] Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., and Nevill-Manning, C.G. (1999) “Kea: Practical automatic keyphrase extraction.” *Proc ACM Conference on Digital Libraries*, Berkeley, CA: ACM Press, 254-255.

[6] *Agrovoc Multilingual Agricultural Thesaurus* (1995) Food and Agricultural Organization of the United Nations.

[7] Rolling, L. (1981) “Indexing consistency, quality and efficiency.” *Information Processing and Management*, 17, 69-76.

[8] Barzilay, R. and Elhadad, M. (1997) “Using lexical chains for text summarization.” *Proc Intelligent Scalable Text Summarization Workshop*, Association for Computational Linguistics.

[9] Reeve, L., Han, H. and Brooks, A.D. (2006) “BioChain: Using lexical chaining methods for biomedical text summarization.” *Proc ACM Symposium on Applied Computing*, Bioinformatics track.

[10] Chali, Y. (2001) “Topic detection using lexical chains.” *Proc Conf on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*. Lecture Notes in Computer Science 2070, Springer.

[11] Morris, J. and Graeme H. (1991) “Lexical cohesion computed by thesaural relations as an indicator of the structure of text.” *Computational Linguistics* 17(1).

[12] Fellbaum, C. (1998) *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

[13] M. Galley and K. McKeown (2003) “Improving word sense disambiguation in lexical chaining.” *Proc Int Joint Conference on Artificial Intelligence*, pp. 1486-1488.

[14] Crane, D., Pascarello E. and James, D. (2005) *Ajax in Action*. Manning, Connecticut

[15] Hearst, M. A. and Pedersen, J. O. (1996) “Visualizing information retrieval results: A demonstration of the TileBar interface.” *Proc Conf on Human Factors in Computing Systems*, pp. 394-395

[16] Denning, P., Horning, J., Parnas, D. and Weinstein, L. (2005) “Wikipedia risks.” *Communications of the ACM* 48(12).