Enhanced Document Retrieval and Discovery Based on a Combination of Implicit and Explicit Document Relationships

Ahmed A.O. Tayeh
WISE Lab
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels, Belgium
atayeh@vub.be

Ngoc Tran
Swinburne University of Technology
Melbourne, Australia
ngoctran@swin.edu.au

Beat Signer
WISE Lab
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels, Belgium
bsigner@vub.be

ABSTRACT

With the rapid increase of digital information we are dealing with in our daily work, we face significant document retrieval and discovery challenges. We present a novel document retrieval and discovery framework that addresses some of the limitations of existing solutions. An innovative aspect of our solution is the combination of implicit and explicit links between documents in the retrieval as well as in the visualisation process, in order to improve document retrieval and discovery. Our framework exploits implicit relationships between documents—defined by the similarity of their content as well as their metadata—and explicit links (hyperlinks) defined between documents based on a third-party link service. Further, the software framework can be extended with arbitrary third-party visualisations. Last but not least, our search query interface offers advanced features not available in most existing document retrieval systems.

CCS CONCEPTS

- **Information systems → Search interfaces**; **Information extraction**; **Clustering and classification**; **Link and co-citation analysis**.

KEYWORDS

Document retrieval; document discovery; search interfaces, document linking; clustering; hypertext; link navigation

ACM Reference Format:
Ahmed A.O. Tayeh, Ngoc Tran, and Beat Signer. 2019. Enhanced Document Retrieval and Discovery Based on a Combination of Implicit and Explicit Document Relationships. In *ACM Symposium on Document Engineering 2019 (DocEng '19), September 23–26, 2019, Berlin, Germany.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3342558.3345407

1 INTRODUCTION

The most common tool we use for managing our documents is still the traditional file browser with the traversal of nested folders as the standard way of locating documents. While this method is the most commonly used one, previous research has shown that it is ineffective and cognitively demanding [6]. An alternative approach is to use a desktop search engine to retrieve documents directly via specific search queries. With this method, the user has the flexibility to search for documents by title or content matching. However, the content matching search mechanism is limited in the sense that in most systems it only returns documents that contain the exact keywords. A simple keyword-based search therefore often misses documents that are relevant for a given search query. Another issue with search-based approaches is that they do not take into account the similarity of documents to enhance document retrieval and discovery. Alternative solutions with novel user interfaces such as DeFiBro [10] and Flip Zooming [2] have been proposed, but most of them have some shortcomings. They either ignore the content of a document, stick with traditional hierarchical tree representations, or only offer a predefined visualisation.

Besides the aforementioned approaches for document discovery, there are a number of studies and systems working towards exploiting document metadata in order to enhance document discovery and retrieval. Dumais et al. [4] have presented a system where users can not only search for objects with words forming part of the metadata but also based on word stems. SurVis [1] and iVisClustering [8] have exploited document metadata and provided differnet interactive visualisations. A last category of systems work towards the vision of the Memex, including Haystack [7], SEMEX [3] and iMapping [5].

We do believe that an enhanced solution for document discovery and retrieval should not neglect the fact that documents do not exist in isolation but are related to other documents. These relationships might be explicitly established via references (e.g. hyperlinks or footnotes) or be implicitly derived based on the similarity of content or some metadata. There has been various research exploiting implicit links between documents to enhance document retrieval and discovery tasks [1, 10]. Nevertheless, to the best of our knowledge, there has been no solution exploiting the combination of explicit and implicit links to enhance document retrieval and discovery. In our proposed approach for document retrieval

DocEng '19, September 23–26, 2019, Berlin, Germany

ACM ISBN 978-1-4503-6887-2/19/09. . . $15.00
https://doi.org/10.1145/3342558.3345407

and discovery, *we exploit the combination of implicit and explicit links between documents not only in the retrieval process but also in the visualisation process.* The proposed solution relies on a clustering algorithm for discovering implicit links between document content and metadata. In addition, our solution exploits explicit links (i.e. bidirectional hyperlinks) that are defined between documents based on an existing link service [12–14].

While the idea of combining implicit and explicit links promises better support for retrieving and discovering documents, it raises a number of questions: How should we visualise both explicit and implicit links in a single interface? How can we deal with different user preferences? Both of these issues stem from the fact that we lack a framework that does not only support the visualisation of both kinds of links but also is flexible enough to support different visualisations.

2 ENHANCED DOCUMENT RETRIEVAL AND DISCOVERY

The strength of the presented framework is its exploitation of the combination of implicit and explicit links between documents. In contrast to some previous research that only used document metadata to discover the implicit links between documents [10], we are employing both a document's content and metadata in order to discover implicit links based on a clustering algorithm. Moreover, we make use of explicit document links that have been created by users of the cross-document link service [12–14]. It is worth mentioning that we also exploited a document's content and metadata to support important features such as synonym-based search and word stems. Further, our framework offers an extensible architecture that supports multiple visualisations. In the remainder of this section, we clarify how document metadata and content as well as the explicit links have been exploited. We further elaborate on the system architecture and present a scenario of using the framework for searching documents, before providing some details about the implementation.

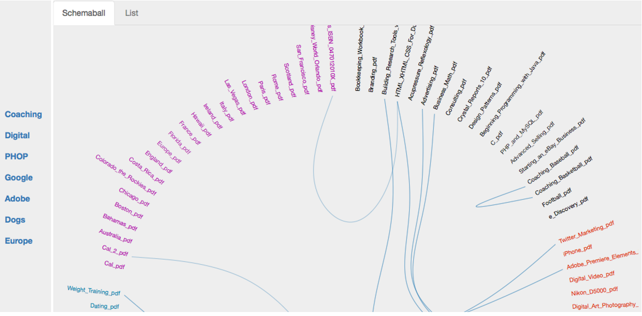


Figure 1: Schemaball visualisation of a search result

We decided to take the document content into account since it can help when a document's metadata causes some confusion or misunderstanding (e.g. based on an incomplete document name or a wrong author name). Document metadata is also taken into account since it is a valuable resource of information that can definitely contribute to the effectiveness and efficiency of any document retrieval system. With the exploitation of metadata, one can provide different perspectives for a document collection and we can, for example, visualise the documents based on their title in combination with the author.

In our framework, a document's content and metadata serve three main purposes. First, they are used to build an inverted index to support full text search. The second purpose is to discover implicit links between different documents via a clustering mechanism to group similar documents in the same category (cluster). Figure 1 illustrates how our approach can visualise documents in distinct clusters with different colours for each cluster. Note that we preferred the dynamic (query-based) clustering over a classic static clustering approach. In static clustering, all the documents (files) are clustered once or whenever a predefined set of new documents are stored in the file system. Therefore, users will always be shown the same document clusters. On the other hand, in dynamic clustering, the cluster analysis is restricted to only the subset of the document collection forming part of the search query result. A search query for a document in our framework will normally return a number of documents partitioned into two subcollections. The first subcollection consists of documents that contain the exact search keyword, its stems and/or its synonyms. The second subcollection contains documents that have explicit links with any document in the first subcollection. In order to enhance the effectiveness of the visualisation and to reduce time and effort [11], we use the dynamic clustering of the document collection returned by an individual search query.

Further, we support search based on synonym matches. By doing so, not only documents that contain the search keyword or its stem are returned, but also documents that contain keywords with the same meaning as the original search keyword. In contrast to the approach described by Mosweunyane et al. [10] which supports this feature for a document's metadata only, we support synonym-based search in both the document content as well as metadata. We make use of the WordNet lexical database from which we query a list of synonyms for every search keyword and then execute the search with the generated list of keywords. Note that we give the user the possibility to enable or disable this feature.

As described earlier, our framework makes use of the explicit links defined via a cross-document link service. The explicit links are an excellent resource for enhancing document retrieval and discovery. The existence of an explicit link between two documents A and B literally means that these two documents are related to each other. Therefore, if document A belongs to the results of a given search query, it is likely that document B may also be of interest to the user. We use this assumption to enhance our search results by adding document B to the search result if it is not already forming part of it. According to the cluster hypothesis [9], documents that are similar to each other tend to belong to the same cluster. However, it is possible that document A and B are not similar content-wise and therefore classified in different clusters. Moreover, when visualising the search results, we take into account existing explicit document links and visualise them as connected curves as shown in Figure 1.

2.1 System Architecture and Implementation

The general architecture of our framework consists of the three main components illustrated in Figure 2. The first component is responsible for handling the search queries entered by the user. A second component is in charge of the document clustering. The third component collects the documents returned by a search query and publishes the results in a neutral data representation which can be further processed by a visualisation engine. The core of our framework is implemented in Java, making use of open source libraries such as Apache Tika[1], Apache Lucene[2] and Apache Mahout[3].

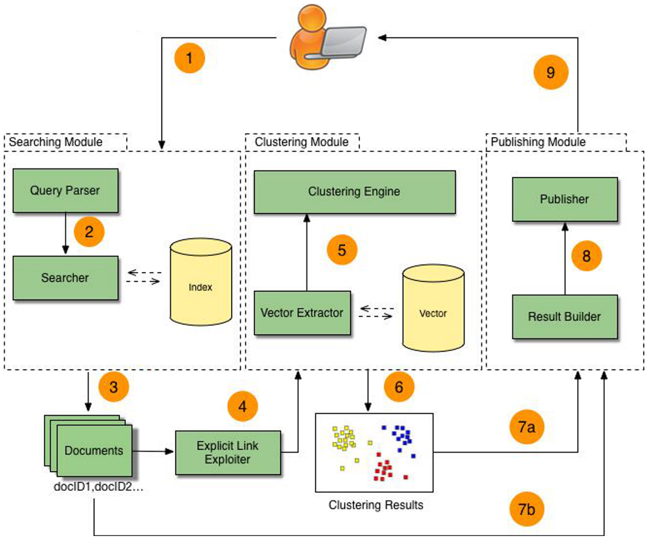
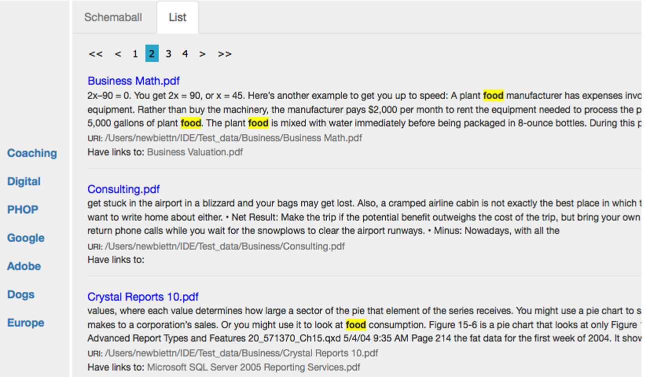


Figure 2: Architecture of the document retrieval framework

In order to illustrate the communication between the different components, we present a scenario (referring to the numbers depicted in Figure 2) of a user who would like to search for a document. The user enters a search query via the search interface (1) enabling them to query documents based on multiple search criteria. A user can search via keyword synonyms or any form of a keyword contained in a document's metadata or content. Furthermore, the user can retrieve documents via an author's name or the date of creation. In contrast to existing systems (e.g. Mac Finder) that enforce a logical AND operator between the different search criteria, we give the user the flexibility to use logical AND and OR operators, as well as a wildcard operator.

The search query is forwarded to the search module in order to be parsed (2). If the user opts for a search using synonyms, the search module will communicate with the WordNet database in order to retrieve all the synonyms of the original keywords (except for the author's name and the date of creation). Both the keywords entered by the user as well as the retrieved synonyms are combined to search for documents in the file system. We take the stems of the search keywords in order to be able to search over the index. The search module will search over the index and a list of matching documents is returned (3). The search result is then passed to the explicit link exploiter engine. As described earlier, the explicit link engine retrieves other documents that are related to documents in the search result (4) by querying explicit link metadata from the cross-document link service. The search result is augmented with these related documents and forwarded to the clustering module. The vector extractor component of the clustering module returns matching vectors based on the list from the vector repository (5). The matching vectors are read and text mining is performed in order to cluster the documents (6). The documents are clustered by using the k-means algorithm and the number of clusters is determined by using the rule of thumb. When the module finishes its task, it returns the final clustering results. Both the search result (7b) as well as the clustering results (7a) are then passed to the publishing module (8). The publishing module provides a RESTful API which formats the search result as JSON data to be further processed by different visualisation engines.



Business Math.pdf
2x–90 = 0. You get 2x = 90, or x = 45. Here's another example to get you up to speed: A plant **food** manufacturer has expenses invo
equipment. Rather than buy the machinery, the manufacturer pays $2,000 per month to rent the equipment needed to process the p
5,000 gallons of plant **food**. The plant **food** is mixed with water immediately before being packaged in 8-ounce bottles. During this p
URI: /Users/newbiettn/IDE/Test_data/Business/Business Math.pdf
Have links to: Business Valuation.pdf

Consulting.pdf
get stuck in the airport in a blizzard and your bags may get lost. Also, a cramped airline cabin is not exactly the best place in which t
want to write home about either. • Net Result: Make the trip if the potential benefit outweighs the cost of the trip, but bring your own
return phone calls while you wait for the snowplows to clear the airport runways. • Minus: Nowadays, with all the
URI: /Users/newbiettn/IDE/Test_data/Business/Consulting.pdf
Have links to:

Crystal Reports 10.pdf
values, where each value determines how large a sector of the pie that element of the series receives. You might use a pie chart to s
makes to a corporation's sales. Or you might use it to look at **food** consumption. Figure 15-6 is a pie chart that looks at only Figure
Advanced Report Types and Features 20_571370_Ch15.qxd 5/4/04 9:35 AM Page 214 the fat data for the first week of 2004. It shov
URI: /Users/newbiettn/IDE/Test_data/Business/Crystal Reports 10.pdf
Have links to: Microsoft SQL Server 2005 Reporting Services.pdf

Figure 3: Ordinary list visualisation of a search result

The presented framework is extensible to support different visualisations. Every visualisation has its strengths and weaknesses and we did not want to restrict our framework to a single visualisation. Normally each visualisation provides a different perspective of the documents, and hence, users can switch between visualisations according to their needs. The support of multiple visualisations further allows us to investigate how different visualisations might affect the document retrieval task. Currently, we have implemented two different visualisations making use of D3.js[4]; a schemaball visualisation and an ordinary list visualisation. In the schemaball

[1] https://tika.apache.org
[2] https://lucene.apache.org/core/
[3] http://mahout.apache.org
[4] http://d3js.org

visualisation, explicit links are visualised by curved lines between different documents. Users can view information about the documents, including some text snippet that contains the matching results and some metadata by hovering over the document name. Moreover, they can filter documents by clicking on one of the cluster names shown in the left panel. In our second list visualisation illustrated in Figure 3, we can show richer textual information, including the filename, snippets of the document and the target document of an existing explicit link. Please note that in both, the schemaball and list visualisation, we visualise the cluster names in the left panel. When a user clicks on a cluster name, only the documents of the corresponding cluster are highlighted. It is worth mentioning that we name any cluster by using the name of the document that is closest to the centroid.

3 DISCUSSION AND FUTURE WORK

To the best of our knowledge, the presented framework for document retrieval and discovery is the first approach considering implicit and explicit links between documents in the retrieval as well as in the visualisation process. The presented document retrieval framework overcomes some limitations of existing document retrieval systems such as offering only a single visualisation, or neglecting document content, metadata as well as manually created explicit links. In order to address a wide range of end-user preferences, we offer an extensible visualisation engine. As mentioned before, even though we currently offer a web-based visualisation of the search results, any third-party application can benefit from the JSON-formatted search results and provide a new visualisation. Last but not least, we have exploited the rich information extracted from documents (e.g. content, metadata, synonyms or stemming results) in order to provide enhanced search features such as boolean or wildcard queries.

There is no doubt that the synonym-based search might possibly help users in finding documents. However, sometimes WordNet returns fifteen synonyms for a single keyword. In practice, this means that a search is performed over a document's content and metadata with sixteen keywords. Of course, this has an impact on the overall performance, in particular if we have to search over a large collection of documents. It is the main reason for giving the user the possibility to turn off the synonym-based search feature.

We plan to perform a detailed evaluation of the presented framework in order to investigate the efficiency and effectiveness of synonym-based search in document retrieval. Moreover, we will also investigate the scalability of each visualisation. Last but not least, we believe that there is still room for further exploiting explicit document links. Hence, we would like to build a small plug-in for our framework that makes use of the existing implicit links between documents in order to suggest new explicit links to the user. Thereby, users will not have the burden to manually add explicit links, but new explicit links could be dynamically recommended by the framework and automatically added after a user's approval.

4 CONCLUSION

We have presented a framework for enhancing the retrieval and discovery of documents. Based on a clustering algorithm in combination with a cross-document link service, our document retrieval framework exploits implicit as well explicit document relationships in order to improve the retrieval and discovery process. Furthermore, the presented solution offers advanced features such as synonym-based search. While we currently support two different search result visualisations, in the future our software framework might be extended with arbitrary third-party visualisations.

REFERENCES

[1] Fabian Beck, Sebastian Koch, and Daniel Weiskopf. 2016. Visual Analysis and Dissemination of Scientific Literature Collections with SurVis. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016). https://doi.org/10.1109/TVCG.2015.2467757

[2] Staffan Björk. 2000. Hierarchical Flip Zooming: Enabling Parallel Exploration of Hierarchical Visualizations. In *Proceedings of AVI 2000*. Palermo, Italy. https://doi.org/10.1145/345513.345324

[3] Yuhan Cai, Xin L. Dong, Alon Halevy, Jing M. Liu, and Jayant Madhavan. 2005. Personal Information Management with SEMEX. In *Proceedings of SIGMOD 2005*. Baltimore, USA. https://doi.org/10.1145/1066157.1066289

[4] Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. 2003. Stuff I've Seen: A System for Personal Information Retrieval and Re-use. In *Proceedings of SIGIR 2003*. Toronto, Canada. https://doi.org/10.1145/860435.860451

[5] Heiko Haller and Andreas Abecker. 2010. iMapping: A Zooming User Interface Approach for Personal and Semantic Knowledge Management. In *Proceedings of Hypertext 2010*. Toronto, Canada. https://doi.org/10.1145/1810617.1810638

[6] Victor Kaptelinin and Mary Czerwinski. 2007. *Beyond the Desktop: Integrated Digital Work Environments*. MIT Press, Chapter Beyond Lifestreams: The Inevitable Demise of the Desktop Metaphor.

[7] David R Karger, Karun Bakshi, David Huynh, Dennis Quan, and Vineet Sinha. 2005. Haystack: A Customizable General-Purpose Information Management Tool for End Users of Semistructured Data. In *Proceedings of CIDR 2003*. Asilomar, USA.

[8] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. 2012. iVisClustering: An Interactive Visual Document Clustering via Topic Modeling. *Computer Graphics Forum* 31, 3 (2012). https://doi.org/10.1111/j.1467-8659.2012.03108.x

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

[10] Gontlafetse Mosweunyane, Leslie Carr, and Nicholas Gibbins. 2011. A Tag-like, Linked Navigation Approach for Retrieval and Discovery of Desktop Documents. In *Digital Information and Communication Technology and Its Applications*. Springer. https://doi.org/10.1007/978-3-642-22027-2_58

[11] Gerard Salton. 1967. Search Strategy and the Optimization of Retrieval Effectiveness. In *Proceeding of FID-IFIP Conference on Mechanized Information Storage, Retrieval and Dissemination*. Amsterdam, The Netherlands.

[12] Beat Signer and Moira C. Norrie. 2007. As We May Link: A General Metamodel for Hypermedia Systems. In *Proceedings of ER 2007*. Auckland, New Zealand. https://doi.org/10.1007/978-3-540-75563-0_25

[13] Ahmed A.O Tayeh, Payam Ebrahimi, and Beat Signer. 2018. Cross-Media Document Linking and Navigation. In *Proceedings of DocEng 2018*. Halifax, Canada. https://doi.org/10.1145/3209280.3209529

[14] Ahmed A.O Tayeh and Beat Signer. 2015. A Dynamically Extensible Open Cross-Document Link Service. In *Proceedings of WISE 2015*. Miami, USA. https://doi.org/10.1007/978-3-319-26190-4_5