

Docco: Document Retrieval with Formal Concept Analysis

Peter Becker¹

School of Information Technology and Electrical Engineering (ITEE)
The University of Queensland
QLD 4072, Australia
`peter@peterbecker.de`

Abstract. A large amount of information available in digital form is not stored in a standardized structure such as a relational database but available only in form of semi-structured documents in a file system. Typically these documents span a number of different file formats and do not follow standard schemas in their structure. With DOCCO we present a tool that allows retrieving documents from a heterogenous collection using Formal Concept Analysis as a method of structuring query results.

1 Introduction

With the rise of desktop computing and modern office applications such as word processors, spreadsheets and presentation software the number of digital documents created has increased to an extent where managing collections of such documents becomes a major issue in work environments and the current approaches of organizing documents in folders in a file system becomes too limited.

As an example imagine a paper presented in a workshop on a conference. Was it classified as belonging to the conference or the workshop or both? Were acronyms or full names used for the conference or workshop? Was the paper also classified against author(s), year or other facets such as the topic of the paper? How is a paper that has multiple authors classified? What does the person retrieving the document do if the file system wants them to choose the conference first, but they know only author and year?

As a new approach to this problem we implemented DOCCO ¹. We used a standard indexing engine and indexed not only the content of the documents, but also embedded metadata (if possible) and the classification made through the position in the file system. Formal Concept Analysis (FCA) is used to visualize the results in a way that allows easy handling of overspecified queries.

2 Connecting Information Retrieval and Formal Concept Analysis

The set of all documents, the set of all words in them (typically in stemmed form) and the ‘document contains word’ relationship is an obvious candidate

¹ <http://tackit.sourceforge.net/docco/>

for a formal context in a system as the one proposed here. The choice taken for DOCCO is slightly different: DOCCO does not operate on a single context, but every combination of query parts creates a new context. Whenever the user enters a query it is broken into smaller parts (single words in the simple case), each part is then queried separately and the results are combined.

The set of objects G in this approach is not the whole document collection, but only those documents which match at least one of the query parts. The attributes M are the query parts, their results the extents of the attribute concepts (m', m'') for $m \in M$. From this, DOCCO calculates the intersections and thus the other extents required for the lattice (the top extent being G itself).

A core advantage of this approach is that G is typically much smaller than the set of all documents, which means calculations are faster. It also fits naturally with the features of a search engine which is specialized in retrieving the documents for a query.

WORD	search for documents containing the word in their body
WORD ₁ WORD ₂	search for documents containing either word in their body
FIELD:WORD	search for documents containing the word in the specified field
"PHRASE"	search for documents containing a phrase in the body
CLAUSE ₁ AND CLAUSE ₂	search for documents fulfilling both query parts
CLAUSE ₁ NOT CLAUSE ₂	search for documents matching the first but not the second part
(QUERY)	can be used to group queries to be used in boolean expressions
FIELD:(QUERY)	applies the query to the field
? *	can be used as wildcards (although not as the first character)
~	can be appended to a word to allow fuzzy matching
+	can be put in front of a clause in a disjunction to make it mandatory
\	can be used to escape special characters

Table 1. LUCENE's query syntax

The search engine used for DOCCO is LUCENE from the Apache project². LUCENE is a search engine written in Java which allows storing the mapping between a set of objects ('documents') to keywords in a way that is very efficient for querying. It also comes with a query system that allows a number of operations to define more complex queries. Table 1 shows an overview of the operations allowed.

title	the document title
name	the file name
author	the author(s) of the document
keywords	keywords used as explicit markup in the metadata
mod_date	date of last modification
creation_date	date of creation
ext	the extension of the file (such as "pdf" or "doc")
path	the full file system path to the document
path_words	the file system path broken down into words
size	the size of the document in the file system

Table 2. Query fields available in DOCCO

² <http://jakarta.apache.org/lucene/>

The fields available are application specific. DOCCO stores a number of fields extracted from the metadata of the documents (such as HTML <META> tags) and from the file system. A list of all fields docco offers is shown in Table 2.

The amount of information available in these fields depends on the document type. DOCCO is able to index a number of document types, the basic installation supports extracting information from plain text, HTML, XML, Open Office and RTF files. Plugins can be installed to support Word, Excel and PDF documents.

3 The User Interface

The main user interface of DOCCO is shown in Figure 1. It contains four central elements: a text field to enter and submit queries; a diagram area in which the lattices are displayed; a tree view to shown result sets in structured form; and a summary area to show details about a single document.

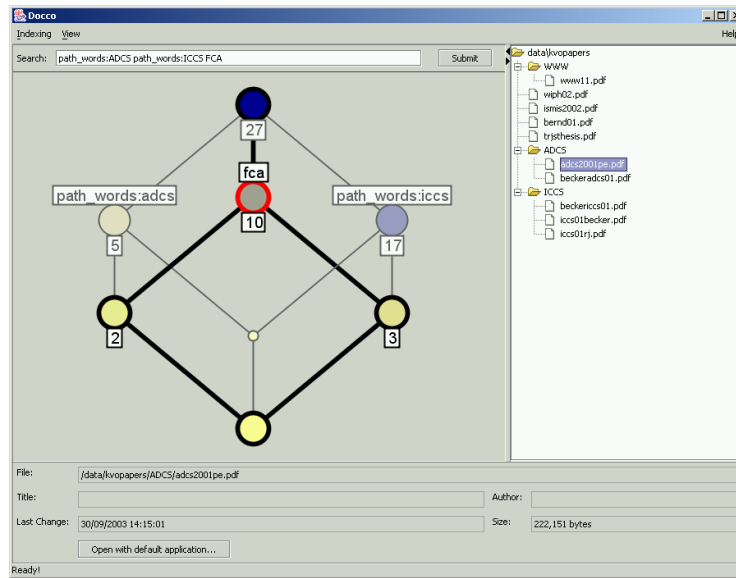


Fig. 1. DOCCO's user interface

To use DOCCO, the user submits a LUCENE query through the query field. DOCCO will then parse the query and split it into the toplevel clauses, for example the query "A (B AND C)" would be split into the two parts "A" and "B AND C". These are then queried separately on the LUCENE index and the results are combined into a lattice which is displayed as a diagram.

The diagram can display the lattice either by itself or embedded into a boolean lattice. Nodes can be dragged around in two ways: standard dragging

with the left mouse button moves the node in a way that the diagram will stay attribute-additive, potentially moving other nodes to ensure this constraint. If the shift key is held while dragging only the node itself is moved.

Selecting a node in the diagram (as shown in Figure 1) shows the extent of the corresponding concept in the tree view to the right. This tree view has two additional features to make it easier to use: it combines multiple levels of directories if the intermediate directories are empty (as seen in the top node of the tree view), and it automatically opens the top levels of the tree view to allow a quick overview.

When the user selects a document from the tree view, its details are displayed in the bottom area. Clicking the button there or double-clicking the document in the tree view opens the document using the standard application.

Figure 1 also shows a particular feature of DOCCO: the words in the file system parts can be used in a query. This means the classification created by filing the document in a particular folder can be reused in DOCCO. In this example the classification against two different conferences was used — a piece of information that would not be available without looking at the file system path.

4 Conclusion and Outlook

DOCCO has been successful in finding a small user base. The visualization using FCA has been welcomed by many users, including people without prior knowledge of the technology.

There are still a number of technical problems, such as the indexing of different document formats or finding a cross-platform approach of opening documents (there is no standard notion of a default application on UNIX systems).

One of the most interesting academic problems is the question of how a program like DOCCO can help the user in expanding a query. Finding suitable keywords to add to a diagram can be quite challenging and is a different problem from expanding a query in a system where overspecified queries are problematic.

Further detail about DOCCO is presented in [1], including references to similar work in the area of email collections.

References

1. Becker, P., Cole, R.: Querying and analysing document collections with formal concept analysis. In: Proceedings of the 8th Australasian Document Computing Symposium, Canberra (2003)