Informer

Number 17, Winter 2005/6
ISSN 0950-4974

BCS INFORMATION RETRIEVAL

In This Issue

About Informer

Informer is the quarterly newsletter of the BCS Information Retrieval Specialist Group (IRSG). It is distributed free to all members. The IRSG is free to join via the BCS website (http://irsg.bcs.org/), which provides access to further IR articles, events and resources.

The British Computer Society (BCS) is the industry body for IT professionals. With members in over 100 countries around the world, the BCS is the leading professional and learned society in the field of computers and information systems.

Informer is best read in printed form. Please feel free to circulate this newsletter among your colleagues.

Welcome to the Winter edition of Informer. It's been a year now since I took on the Editorship of this newsletter, so I'd like to take the opportunity to reflect on some of the changes we've seen over the past twelve months.

First, membership. If you are new to the IRSG, welcome - you are in good company. Membership numbers have more than trebled over the past year, mainly through reaching out to audiences that the IRSG hadn't traditionally focussed on - namely, the thousands of people in the BCS and further afield who have a professional (or indeed personal) interest in information search/retrieval, but aren't necessarily academics or researchers.

Second, community. This is an area where I personally think we could do better. Sure, we have ECIR, which is a great annual forum for IR researchers. And we have Informer, which continues to provide quality, topical IR articles, mainly thanks to the dedication and energy of the members. And we now have Industry Day (see below), which promises to provide a networking and awareness forum for solution vendors, end users and related search professionals.

But can we really claim to have a community? If so, where is the infrastructure? When the IRSG began, having an annual conference and a mailing list was impressive. But times have changed. These days, the barriers to entry are extremely low - you can create on online community in a few clicks, just using something like Yahoo Groups. Moreover, you can then build on that by providing things like member profiles, shared resources, online polls, searchable archives, and lots more. Sure, we still get a few messages posted on ir@jiscmail.ac.uk (mainly conferences announcements and so on). But that's not what I call a community, where members engage in open discussion to share knowledge and contribute to the solution of each other's real-world problems and questions. For an example of what's possible, take a look and some comparable IR-related communities such as KnowledgeBoard, or the FreePint Bar, or TaxoCoP. I certainly think we could learn a lot from the way these forums operate, and this is an issue I'd like to return to, perhaps later in the year. But for now, I think we should just at least be aware of the possibilities.

Finally, events. As you've probably noticed by now, on April 13 we're running Industry Day 2006 (see page 14). This idea for this event was initially floated around a year ago, and I'm pleased to report that during the last 12 months we have secured a great venue and an excellent line up of speakers, including Google, BT, Microsoft, FAST, and many more. Attendance will be limited, so be sure to book your place in advance via the ECIR website. Next edition of Informer will be in April, around the time of this event, so we hope to run a special "Industry Day" issue. Until then, keep sending in those articles and all the valuable feedback.

Best regards,
Tony Rose
Informer Editor and Vice chair, IRSG
Email: irsg@bcs.org.uk

Product Review: Corpora Software

By Alex Bailey

Late last year I was very fortunate to be able to sit down and have a chat with David Phillips, Head of Search Technologies at the Guildford-based Corpora Software, who offer a range of software solutions for enterprise search. We talked over some of the offerings available, and the following is a round-up of the discussion.

Corpora was created in 2000 to develop natural language processing based knowledge discovery and knowledge sharing tools, although the history of some of the products and technology stretch back further due to acquisition of other established enterprise search software providers. The result is a comprehensive portfolio of products that cover many aspects of enterprise knowledge management.

The first product we looked at was Find!, which is Corpora's Enterprise Search Engine. Typically a search engine is the backbone to any enterprise knowledge management system and it's important to get this right. David took me through the features of Find!

Find!

This is a fully-fledged enterprise search engine, with Boolean and natural language queries. It provides security built into the core engine, and a 'cellular architecture' that allows several PCs to each manage an index, resulting in benefits in scalability and redundancy. Dynamic categorisation of documents can be used to filter the search results. And of course, it supports the all the various file formats you are likely find these days.

What struck me throughout the conversation with David was the configurability of the system. An experienced administrator can select multiple indexing strategies, and even chose the level of tokenisation and stemming. There are also multiple search strategies that can be selected to suit a particular search. But most importantly there is the option to adapt the search behaviour to your own needs using trainable agents. These agents can be given feedback to make sure that the right documents are pushed up the search ranking for the right queries. In the wider search arena this is an area where Google is famous, or infamous, for claiming to not interfere with the rankings, and for good reason because they have a duty to remain impartial, even if that duty is somewhat self-imposed. However in the case of Corpora their duty is to their customers, and it's good to see that they empower them with the ability to 'tweak' the ranking to get the most accurate results. Different organisations use terms and documents in different ways, and it's important to be able to let them optimise their search for their own needs.

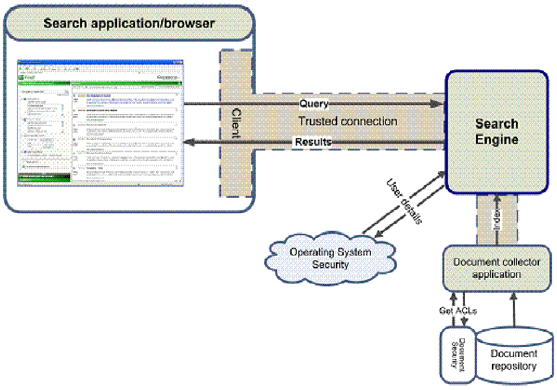
Talk of customisation reminded me of an on-going squabble between Autonomy and Verity, two long-time enterprise search rivals, concerning just how a user should be able to customise the search results. At a trade exhibition some time ago I spoke to Autonomy who were boasting that their probabilistic ranking and configuration technology allowed a more intelligent and accurate way to customise the search through trainable agents. The problem that they faced though was that the underlying parameters were incomprehensible to most ordinary humans and it was almost impossible to get 'under the bonnet' of the system. Verity were well aware of this and they told me that using their human readable rule-based technology to customise the ranking gave the user full control right down to the detail of each rule. However their problem was that it gets pretty hard to maintain large rule-sets and an administrator can end up getting swamped with rules and then can't see the wood for the trees.

I put this to David and asked him how Corpora approached the problem. He told me that the engine uses the latest probabilistic ranking techniques and offers retraining using both positive and negative feedback, and also offers individual term skewing. This allows the administrator to promote or exclude the ranking of specific documents. And specifically to give the user as much control as possible there was an unlimited undo facility, allowing any change of mind to be retraced at any point in time. And if you really wanted, the parameters can be modified by hand, but this was rarely needed.

I also asked if all the customisation options available made the product hard to deploy. In fact the Find! engine can be deployed out of the box in its standard configuration to give a fully working search engine. The interface is implemented as one might expect through a browser, and the back end is supported by an SQL database. The database is used efficiently as a high-level storage medium, which then has the advantage that many company's existing backup and DB administration policies can be applied to the indexes as well. If this isn't required a file-based backend is also available. Corpora can then provide training and consultancy on how the search engine can be set up and further configured.

If a client wants to adapt the search engine further, or integrate it closely with an existing infrastructure, then the low-level APIs are available to create a bespoke system. This does not compromise security at any point because the security is built into the engine at the core level, and neither the browser nor the APIs can be used to circumvent the security in any way.

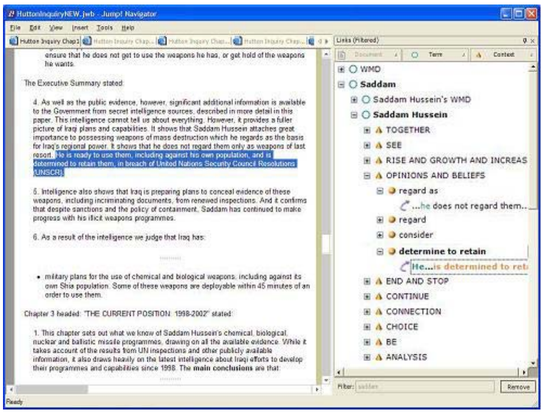
Find! Integrated Security Model



David had previously stated that accuracy and security were top of their customers' requirements, and these had been specifically built into the core of the engine. In fact from this core Find! represents the foundation technology on which much of the product portfolio rests. As you would expect the Find! Engine integrates nicely with the rest of the product portfolio, and two offerings that were of particular interest were the Jump! and Sentiment> products.

Jump!

Jump! is a document navigation tool, launched in 2002, that allows users to browse through documents via information that had been extracted directly from the text. This means that the user can navigate the document according to the people, places, and things mentioned in the document without having to scan the text itself. What is particularly handy is that this is not just a 'search within document' feature because everything is pre-extracted and presented to the user in a hierarchical interface displayed to the right of the document view.



ensure that he does not get to use the weapons he has, or get hold of the weapons he wants.

The Executive Summary stated:

4. As well as the public evidence, however, significant additional information is available to the Government from secret intelligence sources, described in more detail in this paper. This intelligence cannot tell us about everything. However, it provides a fuller picture of Iraqi plans and capabilities. It shows that Saddam Hussein attaches great importance to possessing weapons of mass destruction which he regards as the basis for Iraq's regional power. It shows that he does not regard them only as weapons of last resort. He is ready to use them, including against his own population, and is determined to retain them, in breach of United Nations Security Council Resolutions (UNSCR).

5. Intelligence also shows that Iraq is preparing plans to conceal evidence of these weapons, including incriminating documents, from renewed inspections. And it confirms that despite sanctions and the policy of containment, Saddam has continued to make progress with his illicit weapons programmes.

6. As a result of the intelligence we judge that Iraq has:

- military plans for the use of chemical and biological weapons, including against its own Shia population. Some of these weapons are deployable within 45 minutes of an order to use them.

Chapter 3 headed: "THE CURRENT POSITION: 1998-2002" stated:

1. This chapter sets out what we know of Saddam Hussein's chemical, biological, nuclear and ballistic missile programmes, drawing on all the available evidence. While it takes account of the results from UN inspections and other publicly available information, it also draws heavily on the latest intelligence about Iraqi efforts to develop their programmes and capabilities since 1998. The **main conclusions** are that:

Jump! uses natural language processing to parse the document contents and break sentences down into relationships between nouns and verbs. An index of these entities is created so that the user can select a person, for example 'President Bush', and then select an action, for example 'Communication', and be shown a list of extracts from the text where Bush 'announces', 'states', 'denies', etc. Clicking on an extract then takes you directly to the place in the document from where the extract was taken.

This can be applied not only to single documents, but also larger collections. A typical usage would be to feed the results from the Find! engine into the Jump! navigator which results in a particularly powerful set of tools to pinpoint the exact relevant information without having to trawl through any unnecessary text.

Jump! has received a lot of interest in the US especially from the defence and intelligence community, and very recently last year Corpora was awarded a subcontract to support G&H International Services, Inc. as the Prime Contractor on efforts serving the US Department of Homeland Security, Science and Technology Directorate.

Sentiment

A more recent offering from Corpora that David showed me is the Sentiment product, which analyses news articles with the intention of showing whether the article is positive, neutral, or negative. Combined with a web search engine, or some other news feed, this can be used by a company to get a feel for how it, or its products, are being received in the marketplace. This would then allow a company to pre-empt customer feedback and react accordingly. Automatic sentiment analysis in general is a very recent development in the knowledge management arena and it's encouraging to see this as a commercial offering. Here Corpora is leveraging its experience in natural language processing technology to provide a cutting-edge application.

Conclusion

From the meeting it was evident to me that Corpora have made significant efforts to both develop and acquire a comprehensive suite of search and natural language processing technologies to be able to offer a complete portfolio of knowledge management products. These products can be built up in layers with the search engine providing the foundation, and then adding more levels of sophistication through intelligent application of computation linguistics. Each product is motivated by a genuine business need, and also the company takes issues such as accuracy, security, and systems integration very seriously.

Alex Bailey is experienced in information retrieval and natural language technologies underlying today's knowledge management products. He worked for 5 years at Canon Research Centre Europe investigating and developing document clustering, classification, *summarisation, and information extraction technologies. He is now working as an independent consultant fostering links between industry and academia. Alex is also the One-Day Event Co-ordinator for the IRSG. He can be contacted via: alex.bailey@bcs.org.uk*

David Phillips will be presenting Corpora's knowledge management technology at the IRSG Industry Day in London on the 13th April.

Former BCS-IRSG Chair recognised at Women & Technology Awards 2005

By Fiona Walsh

Ayse Göker, former Chair of the BCS-IRSG and Reader at the Robert Gordon University, was a finalist in the *Best Woman in Technology (Academia)* category at the inaugural *Women and Technology Awards 2005*.

The Women & Technology Awards (UK) recognises excellence and outstanding contributions to technology made by women and the innovative way they use that technology for career and/or business growth. They are open to corporate, public sector, academic and entrepreneurial women, as well as to organisations of all sizes, both from within and outside the technology sector.

At the awards this year, there were eight categories, each one had a strict set of criteria based on elements such as being an inspiration to other women involved in technology, managing a technology roll-out etc. An application and its support were assessed against these criteria by a panel drawn from representatives of BlackBerry, Aurora, sponsors, industry leaders and the business press.

The Awards (UK) were presented at a prestigious dinner ceremony at the Riverbank Plaza Hotel in London in October. For full details of finalists see: http://www.womentechawards.com/finalists.asp

Forthcoming Events

Edited By Andy MacFarlane

Seventeenth Australasian Database Conference (ADC 2006)
Hobart, Australia. 16-19 January 2006. A general database conference with a theme on information retrieval.
https://www.se.auckland.ac.nz/~adc06/

Seventh International Association for Pattern Recognition Workshop on Document Analysis Systems (DAS 2006)
Nelson, New Zealand, 13 - 15 February 2006. A general workshop on document analysis which has various themes of interest including document image retrieval systems.
http://ww.iam.unibe.ch/das06

6th Dutch-Belgian Information Retrieval Workshop (DIR'06)
TNO ICT, Delft, The Netherlands. 13th – 14th March 2006. An annual Dutch/Belgian IR workshop.
http://hmi.ewi.utwente.nl/conference/dir2006

11th Conference of the European Chapter of the Association for computational Linguistics (EACL'06)
Trento, Italy. 3-7 April 2006. A computational linguistics conference which also has a theme on information retrieval. http://eacl06.itc.it/

Adaptive Text Extraction and Mining (ATEM 2006)
Trento, Italy, 4th April 2006, A general text mining workshop with a theme on information retrieval (part of 11th Conference of the European Chapter of the Association for Computational Linguistics). http://tcc.itc.it/events/atem2006/

European Conference on Information Retrieval (ECIR 2006)
London, England, UK. 10th – 12th April 2006. The leading European Information Retrieval Conference, held this year at Imperial College London. http://ecir2006.soi.city.ac.uk/

BCS IRSG Industry Day
London, England, UK. 13th April 2006. A one day event devoted to the challenges involved in designing and developing operational IR products and services. This event follows ECIR 2006.
http://ecir2006.soi.city.ac.uk/index.php?page=indust

Text Mining 2006
Hyatt Regency, Bethesda, Maryland, 22nd April 2006. A text mining workshop with a theme on IR (held in conjunction with the Sixth SIAM International Conference on Data Mining). http://www.cs.utk.edu/tmw06/

Special Track on Information Access and retrieval, 2006 ACM Symposium on Applied Computing (SAC 2006)
Dijon, France, 23-27 April 2006. A conference with a more applied focus on information retrieval. http://www.cis.strath.ac.uk/external/SAC2006/

Search Engine Meeting 2006
Boston, U.S.A. 24th – 25th April 2006. The annual search engine meeting, attended by all the major players.
http://www.infonortics.com/searchengines/sh06/06pro.html

Language Resources for Content-Based Image Retrieval Workshop (OntoImage 2006)
Genoa, Italy, 22nd May 2006. A workshop on creating vocabularies for image retrieval (in conjunction with LREC 2006).
http://www.lrec-conf.org/lrec2006/IMG/pdf/LRECworkshopOntoImage.htm

15th International World Wide Web Conference (WWW 2006)
Edinburgh, UK, 22nd –26th May 2006, A special conference track devoted to Web Search. http://www2006.org/tracks/search.php

Workshop on Pattern Recognition in Information Systems (PRIS-2006)
Paphos, Cyprus, 23rd – 24th May 2006. A general workshop on pattern recognition, with a theme on IR. The workshop is co-located with the International Conference on Enterprise Information Systems (ICEIS).
http://www.iceis.org/workshops_list.htm#PRIS

Libraries in the Digital Age (LIDA 2006)
Dubrovnik and Mljet, Croatia. 29 May - 4 June 2006. An annual course and conference on Digital Libraries. http://www.ffos.hr/lida/

1st International conference on scalable information systems (INFOSCALE 2006).
Hong Kong, 30 May – 1 June 2006. A conference devoted to the issue of scability in information systems, with a theme on information retrieval. http://www.infoscale.org/

Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL 2006)
New York Marriott at the Brooklyn Bridge, Brooklyn, New York, 4th – 9th June 2006. A conference that will be of interest to members who work in the area of Computational Linguistics and IR.
http://nlp.cs.nyu.edu/hlt-naacl06/

Seventh International Conference on Flexible Query Answering Systems (FQAS 2006)
Università degli Studi di Milano Bicocca, Milano, Italy. 7th to 10th June 2006. a query answering conference with a number of themes of interest to IR researchers and practitioners.
http://fqas2006.disco.unimib.it/

Joint Conference on Digital Libraries: Opening Information Horizons (JCDL 2006)
Chapel Hill, North Carolina, USA, 11th –15th June 2006. A digital library conference that will be of interest to members working on search in such systems. http://www.jcdl2006.org/

Adaptive Hypermedia and Adaptive Web-Based Systems 2006 (AH'06)
National College of Ireland, Dublin, Ireland, 21st – 23nd June, 2006. A general conference on adaptive web based systems including at theme on IR. http://www.ah2006.org/

2006 ACM SIGMOD International Conference on Management of Data/25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (SIGMOD/PODS 2006)
Chicago, Illinois, U.SA. 26-28 June 2006. Database conference/symposium with a theme on information retrieval.
http://tangra.si.umich.edu/clair/sigmod-pods06/

Book Review:

Information Visualization: Beyond the Horizon, by Chaomei Chen

Reviewed by Terence Clifton

Aimed squarely at researchers and graduate students, Information Visualization: Beyond The Horizon is an ambitious text, attempting to unify the many differing strands of visualization research under a cohesive framework. This is no mean task, and certainly one that is hindered by the length of the text, at only 316 pages. Although a comprehensive amount of material is covered, attacking this volume with limited background knowledge or without a supporting text would be a difficult task. Many of the sections of the book assume some prior knowledge of either the domain in question or, more often than not, many of the fundamental algorithms associated with the particular form of visualization being addressed. Bearing in mind the target audience for this text, however, this is not an unreasonable assumption to make, and allows the author much greater freedom to delve into the technical and social implications of the various applications and scenarios.

The author has a formal, yet pleasant, writing style, which may seem a little dry to some, but ensures that the salient points are presented in the most efficient and concise manner. This does make some of the more technical chapters a little difficult to read in places, but overall ensures a comprehensive study of the state-of-the-art in visualization without the unnecessary chaff that often accompanies this kind of book.

The book is loosely described by the author as being separated into two sections, the first covering the fundamental concepts and techniques employed in the majority of information visualization scenarios, and the second looking more to the future of the field. Structural extraction and graph-drawing algorithms are given over to comprehensive study in the early chapters of the book, and the content here is comprehensive without being ground-breaking. It is in the later chapters that the author really finds his feet, and presents an excellent survey of current systems and practices (although one would expect this to date rapidly in such a fast-moving field). The chapter on empirical studies of Information Visualization gives a much needed grounding to this highly theoretically driven area of research, and addresses many of the concerns echoed in the current literature with respect to research for the sake of research, without a clear goal or scenario where developments can be usefully applied.

The author's attention to cross-discipline implications with the field is borne out through the chapter on Virtual Environments, which betrays the change of focus from the first edition of this book (Information Visualization and Virtual Environments). This is arguably the weakest chapter of the text, and misses a real opportunity to bring together these highly coherent fields, and challenge the advances and direction of one of the 'hot topics' in current CS research. Although the material is thorough in as far as it goes, it seems a little dated, failing to recognise some of the more recent advances in augmented and virtual reality with respect to collaborative environments and visualization.

"where this text really excels is in its focus on application"

The concluding chapter on recognising emerging trends proposes that Information Visualization is in the process of moving away from the structural nature of its early success to a more dynamic paradigm, capable of recognising abrupt changes in larger, more complex information environments. This chapter, in particular, demonstrates the author's expertise and knowledge of the field, and provides an excellent conduit for the ground-breaking ideas driving the forefront of Information Visualization research. Despite the obvious difficulty in formalising the trends and directions of such a fast-moving discipline, this chapter presents a number of worthy ideas and scenarios that will likely drive the current crop of Information Visualization researchers in years to come.

In summary, Information Visualization is a comprehensive text, providing a comprehensive level of information on what is essentially a limitless topic. Although not an easy read, and certainly not for the layman, the text is nevertheless an excellent tool for the researcher to have to hand. Offering a concise overview of the fundamental concepts, where this text really excels is in its focus on application, and extensive referral to 'real-world' examples of Information Visualization, and particularly its use in cross discipline research. If you are a novice in the field, looking for a gentle introduction, look elsewhere. If, however, you know the fundamentals and are looking for a comprehensive survey of the state-of-the-art, essential details on empirical grounding of techniques, and in particular a wealth of application specific information, then this is likely to be the text for you.

Terence Clifton is a PhD research student in the School of Informatics at the University of Wales, Bangor, where, despite his interest and involvement in Information Retrieval and Artificial Intelligence, he is actually studying for a doctorate in the field of Computer Graphics. He is however, and integral member of the Artificial Intelligence and Intelligent Agents research group at Bangor, and one of the main developers on the groups agent-based Question Answering System – QITEKAT. He can be contacted via:
terence@informatics.bangor.ac.uk

Feature Article: Jakarta Lucene

By Urban Bettag

Information retrieval functionality can be spotted in almost every application: e-mail, instant messaging, web browser, desktop search tools, corporate Intranet or everybody's favourite search engine. In order to support sophisticated search functionality requirements, application architects have to face often the same old dilemma: Pick a commercial search package or build the required search functionality from scratch? The latest commercial enterprise search offerings from vendors such as, Autonomy, Google, Convera or FAST are powerful, but pricey too. A shoe string budget and pressuring deadlines are the most common constraints in today's projects; considering open source software for a start seems to be a fair alternative.

Lucene is a Java based Open Source library which provides full-text search and indexing functionality. Instead of an out of the box application, Lucene offers a usable API for programmers and operates on a lower level. Lucene is managed by the Apache Software Foundation (ASF) and is part of the wider Apache Jakarta project. The latest version 1.4.3 can be obtained from the Apache Jakarta web site. Due to its maturity, robustness and wide adoption Lucene has been ported to other languages as well, such as C++, C#/.NET, Perl, Python and Ruby. The Apache license Version 2.0 applies for Lucene. The library can be used for free, source code can be modified, repackaged and deployed with your own application as long as a clear reference to the ASF is made and the license file is included.

First impressions

The latest version of Lucene is available at the Jakarta project web site and comes with comprehensive API documentation, examples, source code and the actual binary package (~

300K Byte). A packaged web version more suitable for web-based systems is included too.

In contrast to other search libraries Lucene does not search file by file. The search space is analysed first and translated into a normalised representation - the index. Its content, for example a list of words and the physical location make up such an index. Lucene uses a reverse index. All words in the index are unique, that means the index is a compressed representation of the search space. Very similar to the index in a book, looking up a keyword or subject will provide the relevant page.

The first step to get started with Lucene is to create such an index and index the files that need to be searched. Lucene holds the content to be indexed together in a *Document* object. The index is populated by adding new documents to the index or sub index (segment). Each document consists of terms, which in itself consist of a number of fields, ie. name/value pairs (see Figure 1). Typical name/value pairs are for example, title, author or only the text. The actual source data can be any directory file system, content of a database or even a remote web site. However, Lucene can only process text.

The actual text can be embedded in different file formats. In order to keep the library small Lucene only supports plain-text files. However, a variety of free open source document parsers are available for document types such as, RTF, PDF, HTML, XML, OpenOffice, MS Word, MS Excel or MS PowerPoint. Like any other proprietary document format can be supported by developing a custom document filter. The filter provides access to the file, distils the text out of the document and adds the text to a Lucene *Document* object.

Depending on the nature of the text content various analysers are on offer. For example, text can be analysed with a white space analyzer which breaks down the text in tokens separated by white space. Another analyser would filter out all common English stop words (eg, that, the, this, etc.). If we have text in various languages then a language (ie, German, French) specific analyser will be required as stop words vary by language.

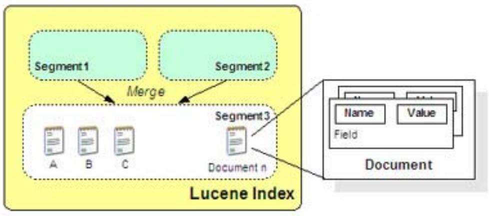


Figure 1 – Lucene Index Structure

To keep the response time short the process (see Figure 2) of generating and optimising the index is separated. *IndexWriter* and *IndexReader* are the responsible classes in Lucene. The *IndexWriter* creates the index and *IndexReader* utilises the index. Even if Lucene is used in multiple applications, *IndexReader* can share the same index and use it concurrently.

How does the indexing work?

The heart of Lucene is the actual *Indexer* class. Depending on the corpus source and document type, the indexer will analyse the source and translates it into a stream of tokens. The tokens will be part of a reverse word list index. Next, the index will be optimised by eliminating all the fill words; this will keep the index small and reduce potential query times. The list of stop words can vary from language, in a multi language search space all languages have to be considered.

In addition the number of co occurrence of a word needs to be considered as well. For example, files in a specific domain usually share the same vocabulary. A small list of words will make up the index and refer to many locations. In order to achieve meaningful search results, the relative and absolute number of occurrence is stored in the index and evaluated in a query.

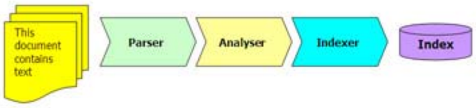


Figure 2 – Lucene Index Process

Compared to the corpus the size of the actual index represents approx. 30%.

Lucene creates for every document a segment and joins them together to keep the number of segments small. Two segments can merge, a new segment will be created and the two others will be removed. The number/size of segments can be configured and tuned for the actual problem. In contrast to other search engines or commercial products, Lucene does not use B-Trees for index optimisation.

The physical index consists of three files. The first file is a directory index and contains offsets, which refer to the directory. The entry in directory refers to the source or physical location. Luke is useful administration tool, which allows browsing through the Lucene index.

countr**ies**	⇨	countri
automat**ion**	⇨	automat
bank**ed**	⇨	bank
bank**s**	⇨	bank

Figure 3: Stemmer

The index gets normalized by applying a stemming and lemmatisation algorithms. The Porter Algorithm breaks away the suffix from a word so that only the stem remains. Like the stop list the stemmer is language dependent. Also, cutting of characters isn't enough for irregular words, e.g. it is not possible to conclude from "went" to "go" by just cutting of characters. A lemmatizer solves these problems, i.e. it always produces real words, even for irregular forms. It usually needs a table of irregular forms for this.

Got a query?

Once the index is in place it can be queried. The query can range from a simple keyword search to a complex search phrase with logical expression. A keyword search in Lucene is established with as a *TermQuery*. In addition, the query can filter and restrict specific fields on the index as well with the +/- operators. Boolean expressions (AND/OR) and even an own search syntax can be established. Fields can be searched with wildcard operators (?,*), range, fuzzy (~) and proximity search.
Lucene provides for every mentioned type its own class. *TermQuery*, *RangeQuery*, *PrefixQuery*, *BooleanQuery*, *WildcardQuery* and *FuzzyQuery* are the relevant classes. More complex search queries, which are based on different types, need to be parsed by the *QueryParser*.

Similar to the indexing process the constructing a query with the Lucene API is fairly simple and requires four separate classes: *IndexSearcher*, *Query*, *QueryParser* and *Hits* class. An instance of the *IndexSearcher* class provides a reference to the Lucene index. *IndexSearcher* passed on a *Query* object, which holds the actual search expression. *QueryParser* does all the more clever things, such as analysing the search string, removing stop words, checking the syntax and retrieving the matching *Documents* out of the index. The *QueryParser* must use the same analyser as during the indexing. A query tree will be established and processed in order to determine the marching document. The result will be determined by a scoring model and collected in a *Hits* object.

```
subject:lucene +query -scoring
```

Figure 4: Query Example

Figure 4 explains the basic search syntax. All documents with the "lucene" mentioned in the title field and "query" mentioned in the text are searched, but those with "scoring" mentioned are omitted.
In case the search syntax is very specific then the query parser can be extended and customised.

Conclusion

Information retrieval requirements are becoming more and more complex as the scope and reach of the information expands. Due the nature of the information, language, domain, document types, taxonomy, visual representation, context mapping, application integration are just a small selection requirements which have to be taken into account when making decisions about application design and considering the right information retrieval kit.

Lucene impresses with its flexibility, features, tools and performance. Very small technology footprint requires only 1 MB RAM on the heap and achieves respectable processing throughputs. The API is very easy to use and it is a joy to work with. No surprise that the number of projects and organisations powered by Lucene is growing fast.

Urban Bettag studied Computer Science at the University of Karlsruhe, Germany. He is a London-based Technology Consultant and has been with Reuters, ABN Amro and DHL in the past. His interests include software architecture, Open Source Software and web-based technologies. In his leisure time you might find Urban running around London training for marathons. If you have any comments or questions regarding Jakarta Lucene, feel free to get in touch with Urban via urban@bettag.com.

Get Involved!

Informer welcomes contributions on any aspect of information retrieval. We are particularly interested in feature articles and opinion pieces, but are also pleased to receive news articles, book reviews, jobs ads, etc.

Right now we are running a series of Product Reviews, so if you are interested in reviewing any of the following:

- Copernic
- Ask Jeeves Desktop Search
- Blinkx
- MSN Search Toolbar

Then please get in touch with us via irsg@bcs.org.uk. All of the above are freely available as software downloads.

Research Update:

Incorporating Context using the Assumptions of Language Models

By Leif Azzopardi

Since their introduction in 1998, Language models applied to searching and retrieval have become very popular and widespread because of their simplicity, intuitiveness and, of course, effectiveness. The basic idea behind the approach is based on sampling query terms from a document, where documents are ranked according to the probability of the query terms being generated from the document (or the probability of a query given the document, known as the query likelihood). Obviously, the more prevalent query terms are in a document, the higher the query likelihood. This is assumed to mean that the document is more relevant to the user's information need.

Consequently, this is one of the overarching assumptions it makes about the retrieval process. Whilst, intuitively appealing, it has attracted some criticism because the very notion of relevance has been ignored, which leads to some serious theoretical problems. Nonetheless, for this first assumption to hold, two other assumptions are required. Despite the numerous models proposed extending or using Language Modelling, the fundamental assumptions of the model have not been explicitly formalized, nor have they been verified in practise! Thus, this forms a major contribution of my work.

The Assumptions

In my PhD thesis, I provide an overview of the main approaches of Language Modelling and surmise the three main assumptions engaged. Briefly the assumptions are:

1. Correlation: That the query likelihood is correlated with the document's relevance (as previously mentioned). Indeed, the Language Modelling approach, because it ignores relevant would seem to assume that all documents are relevant, and that the query likelihood will tell us just how relevant the document is.

2. Unification: That the data model (i.e. how we represent the document, which is defined by the probability of a term occurring in a document) and retrieval model (matching function employed, here the query likelihood) are one and the same. This means that the way we represent our documents directly influences our retrieval model. So if we change our representation we will change our ranking. This has two important implications: (a) if we improve the document's representation then we will improve the retrieval effectiveness, and (b) a further assumption is required, that the user has some idea of the terms used within documents.

3. Discrimination: Finally, that when a user submits a query to the system they will choose query terms that will discriminate relevant from non-relevant documents. Obviously, if a user issues such a query then there will be a strong correlation found in assumption 1, which introduces a circular argument.

Stating the assumptions explicitly means that each aspect can be scrutinized. Consequently, a deeper understanding of the model and its usage can be developed. For instance, from the second assumption it is possible to estimate the parameters of the language model by making the best possible representation of the data. Whereas the third assumption quite emphatically prescribes the types of terms that a user should submit to the retrieval system. In my thesis, I assess each of these assumptions and show when these assumptions hold, and when they do not.

Injecting Context

Another major goal of my PhD work was to explore the "Context Hypothesis", which can be stated as follows:

> *"Semantically related documents tend to be relevant to the same request"*

Obviously, this is related to the Cluster Hypothesis, however the difference is that instead of focusing on relationships defined by a similarity metric. Instead, I concentrate on those associations formed by semantic relationships between documents, which provide the context for that document (i.e. how one document relates to another document prescribe by the user).On the basis of the assumptions, I motivate how context can be naturally embedded when modelling the language used within documents. Specifically, the second assumption requires that the user will have some idea of the kinds of terms that would be in the document they are searching for. By using the semantic associations between documents to represent the user's understanding of a document, we can build a representation that is more in tune with what the user had in mind. This results in building *context based document models.*

Then, the context based document models were evaluated against standard document models (which do not consider context). In a number of scenarios, I evaluate different types of semantic associations between documents (such as web links, topics of interest, and semantic clustering techniques) to determine whether they can be used first to improve the representation of the documents and then secondly to improve the retrieval effectiveness. Whilst both are possible, it is the identification of the conditions when retrieval performance will be improved by using context that proved illusive and remains an open challenge.

Conclusions

Through the course of my thesis, a key goal was to develop a better understanding of the Language Modelling approach. It is difficult to provide a full account here, so I'll defer avid readers to my thesis. Finally, I'd like to express my gratitude to my supervisors Mark Girolami, Keith van Rijsbergen and Malcolm Crowe for their support and encouragement and also to the University of Paisley, Memex Technology Ltd, and the Overseas Research Students Award Scheme for their financial support.

Leif Azzopardi is a Research Fellow at the University of Strathclyde in Glasgow, UK. His research interests include: formal models for information retrieval, distributed information retrieval and evaluation of information access systems. He can be contacted by email via: leif.azzopardi@cis.strath.ac.uk.

Book Review:

Information Retrieval: Algorithms and Heuristics, by David Grossman and Ophir Frieder

Reviewed by Melanie Kendell

To sum up this review in one sentence - this is a book that has great breadth and depth, but neglects some of the shallows. If you are a computer science student with a good lecturer who will expand on the concepts and give step-by-step introductions to complex equations, this is an excellent textbook. It has a comprehensive coverage of different approaches and utilities and gets down and dirty with algorithm details.

The topic structure is logical and the chunking is set at a good level (neither too long nor too short). Headings are relevant to the content and the language used is clear and concise. This makes it a good reference to look up details such as what algorithm you might use in a particular set of circumstances.

Alternatively, if you have a lazy or incompetent lecturer that barely expands on the text (I presume that IR is not immune to this blight on education), you may want to find some supplemental reading material.

The other stated audience for this book is "practitioners who work on search-related applications". Specifically, it is for those that use algorithms to extract information from largely unstructured sources rather than those that use queries to extract information from such sources as semantically rich XML. I fall into the latter category but was interested to learn about the former – but I must say as a beginner, I found it tough going.

It is explicitly stated in the foreword that "A sophisticated mathematical background is not required", but that depends on your definition of the word sophisticated. Although I am not completely maths illiterate (I studied first year A level maths and stats - before discovering the pub next door), assumptions were made about the level of mathematical knowledge that I obviously lack – ooh, they've chucked in a logarithm, I remember learning to look them up in log tables (yes I'm that old) but I couldn't quite remember why you'd use them off the top of my head. An existing in-depth knowledge of vectors would definitely have made things easier.

I also found the authors' eagerness to get into the nitty gritty of the algorithms left me somewhat bemused. Sometimes an explanation followed the presentation of a (to me) arcane algorithm, but it was too late as I was already gibbering in the corner going "what the…".

Given the deadline for this review was fast approaching, I decided to trawl for tips on XML (an area of specific interest to me) but it seems that the updates for this edition didn't make it as far as the index even though the emergence of XML is cited as one of the reasons for releasing a second edition. Presumably only the page numbers were refreshed - there was no XML entry. If I did find all the XML information, it was fairly sparse which was not surprising as it is designed more for query-based searching than algorithmic searching.

"as a beginner, I found it tough going"

There was also a slight problem with some of the figures and tables being slightly too far away from the relevant text for comfort.

What I did like was explanations such as the difference between precision (the relevance of documents in the retrieved set, ignoring that there may be texts that are just as relevant that weren't retrieved) and recall (the number of relevant texts that were retrieved, ignoring that your results may include clutter from lots of less relevant texts) a balance of which gives the level of effectiveness for a particular application.

I was also impressed by the emphasis on efficiency. In these days where users expect almost instantaneous response ("if Google can search the whole web in under a second, why can't I search a dozen databases in less than ten seconds"), it is important to temper a perfect result set with the time taken to achieve it.

For those of you directly involved in the exploration of different approaches to algorithmic search, and with a prior understanding of general concepts, I can see that this book would be invaluable. You may be exposed to approaches that you hadn't previously considered and the lowdown on the advantages and drawbacks of different methods will aid you in the selection process including ways to blend approaches to achieve a better result.

To my mind, a greater emphasis on explaining general concepts and a gentler lead into the algorithms would have made this book appealing to a much broader audience. Maybe the authors would consider this for a third edition.

Melanie Kendell is an Information Management Consultant specialising in multiple media publications, single-sourcing, customisation, and translation of documentation sets. Her fledgling consultancy can offer practical advice on a range of solutions from the simplest strategies to achieving success with XML-based content management systems. She can be contacted via: melaniekendell@emophus.com.au

BCS IRSG Industry Day

in association with
28th European Conference on
Information Retrieval (ECIR 2006)

13th April 2006
BCS HQ, London, UK

For the first time in its history, the IRSG's annual conference (ECIR) will be followed by a special day devoted to the interests and needs of IR practitioners. This forum presents an opportunity for commercial organisations and individuals to share their experiences with a wider audience, and for researchers to learn more about the issues and problems faced by IR practitioners in developing practical solutions for the information search and retrieval industry.

The programme is currently being finalised but at the time of going to press we have confirmed presentations from:

- Google
- FAST Search and Transfer
- BT
- Microsoft
- And many more!

To book your place at this unique event, register via the ECIR website (follow the link for "BCS IRSG Industry Day")

BCS HQ is 10 mins by Tube from the main ECIR conference venue. A separate one-day registration rate will be available.

Contacts

Web:	http://irsg.bcs.org/
Email:	irsg@bcs.org.uk
Subscriptions:	http://irsg.bcs.org/membership.php
ISSN:	0950-4974

To subscribe, unsubscribe, change email address or contact details please visit http://irsg.bcs.org/ or email irsgmembership@bcs.org.uk.

The IRSG is a specialist group of the British Computer Society.
To automatically receive your own copy of Informer, simply join the IRSG via the IRSG website.