UPGRADE is the European Journal for the Informatics Professional, published bimonthly at <http://www.upgrade-cepis.org/>

Publisher
UPGRADE is published on behalf of CEPIS (Council of European Professional Informatics Societies, <http://www.cepis.org/>) by **Novática** <http://www.ati.es/novatica/>, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática,* <http://www.ati.es/>)

UPGRADE monographs are also published in Spanish (full version printed; summary, abstracts and some articles online) by **Novática**

UPGRADE was created in October 2000 by CEPIS and was first published by **Novática** and **INFORMATIK/INFORMATIQUE**, bimonthly journal of SVI/FSI (Swiss Federation of Professional Informatics Societies, <http://www.svifsi.ch/>)

UPGRADE is the anchor point for **UP**ENET (**UP**GRADE European NETwork), the network of CEPIS member societies' publications, that currently includes the following ones:
• **Informatik-Spektrum**, journal published by Springer Verlag on behalf of the CEPIS societies GI, Germany, and SI, Switzerland
• **ITNOW**, magazine published by Oxford University Press on behalf of the British CEPIS society BCS
• **Mondo Digitale**, digital journal from the Italian CEPIS society AICA
• **Novática**, journal from the Spanish CEPIS society ATI
• **OCG Journal**, journal from the Austrian CEPIS society OCG
• **Pliroforiki**, journal from the Cyprus CEPIS society CCS
• **Pro Dialog**, journal from the Polish CEPIS society PTI-PIPS

ISSN 1684-5285

Monograph of next issue (April 2007)
"Information Technologies for Visually Impaired People"
(The full schedule of **UP**GRADE is available at our website)

CEPIS **UP**GRADE
The European Journal for the Informatics Professional
http://www.upgrade-cepis.org
Vol. VIII, issue No. 1, February 2007

Monograph: Next Generation Web Search
(published jointly with Novática*)
Guest Editors: *Ricardo Baeza-Yates, José-María Gómez-Hidalgo, and Paolo Boldi*

UPENET (UPGRADE European NETwork)

CEPIS NEWS

* This monograph will be also published in Spanish (full version printed; summary, abstracts, and some articles online) by **Novática**, journal of the Spanish CEPIS society ATI (*Asociación de Técnicos de Informática*) at <http://www.ati.es/novatica/>.

Research Directions in Terrier: a Search Engine for Advanced Retrieval on the Web

Iadh Ounis, Christina Lioma, Craig Macdonald, and Vassilis Plachouras

This paper describes the Terrier search engine, giving an overview of its architecture and main Information Retrieval (IR) features, and reviewing the cutting-edge research implemented in it, with a special focus on Web search. IR research is concerned with developing and evaluating search engines that retrieve relevant documents in response to a user query. Terrier is a highly flexible, efficient, effective and robust platform for IR research, readily deployable on large-scale collections of documents [10]. Terrier implements state-of-the-art theoretically-founded models for IR, ranging from formal disciplines, such as probability theory, statistics and natural language processing, to computational aspects of index compression and retrieval efficiency. The research put into Terrier constantly expands towards new branches of the wider IR field, making Terrier a strong, modular and state-of-the-art platform for developing and assessing new concepts and ideas.

Keywords: Information Retrieval, Terrier Research Platform, Web Search.

1 Introduction

The aim of this paper is to present an overview of the research in Information Retrieval (IR) that has been implemented in Terrier, with special focus on Web search. Terrier, TERabyte RetrIEveR, is a high performance and scalable search engine that allows for the rapid development of large-scale retrieval applications, by providing a comprehensive, flexible, robust and transparent platform for research and experimentation in IR [10]. A small part of the Terrier retrieval platform is made available periodically as open source software[1] . Terrier was initiated to facilitate research into Web search, but has since been extended to include other applications, such as Desktop and Web corporate search (intranet search).

The task of a search engine is to retrieve relevant documents in response to a user need, often formulated as a query. To do so, search engines use retrieval models to estimate the relevance of documents to user queries, and a variety of retrieval enhancing techniques to improve the ranking of relevant documents that they offer back to the users. The retrieval models and techniques implemented in Terrier include: a novel and highly effective probabilistic framework for weighting models [1], which do not require tuning; several new retrieval methods tailored for Web search [13] [14]; various selective combination of evidence Web retrieval enhancing approaches [13] [15]; a novel syntactically-based information processing approach [4][6], with applications

[1] The latest open source 1.0.2 version of Terrier can be downloaded from: <http://ir.dcs.gla.ac.uk/terrier/download.html>.

Authors

Iadh Ounis is a Reader in the Department of Computing Science at the University of Glasgow. He holds a Ph.D from the University Joseph Fourier, Grenoble. He has been an active researcher in Information Retrieval (IR) since 1994. He has been involved in a number of projects and working groups on IR, is on the program committees of major retrieval information conferences, is a member of the editorial board of the Information Processing & Management journal, and has chaired a number of IR events and initiatives. His current research focuses on parameter-free probabilistic IR Models, Intranet, Enterprise, Blogs, and Web search, as well as large-scale text retrieval systems building and evaluation. He is the principle investigator of Terrier. <ounis@dcs.gla.ac.uk>.

Christina Lioma is a Ph.D candidate with the IR Group, in the Department of Computing Science at the University of Glasgow. She holds a M.A. (Hons) in Languages and Linguistics from the University of Glasgow, and a M.Sc. (with Distinction) in Natural Language Processing from the University of Manchester. Her research is on Computational Linguistics and Natural Language Processing for textual Information Retrieval. <xristina@dcs.gla.ac.uk>.

Craig Macdonald is a Ph.D candidate with the IR Group, in the Department of Computing Science at the University of Glasgow. He holds a B.Sc. (Hons) in Computing Science from the University of Glasgow. His research interests include IR in the Enterprise, Blogosphere and Web settings. <craigm@dcs.gla.ac.uk>.

Vassilis Plachouras holds a B.Eng. from the National Technical University of Athens, Greece, and a Ph.D in Computer Science from the University of Glasgow. During the study towards his Ph.D, he was involved in the development of Terrier. Currently, he is a post-doctoral researcher at Yahoo! Research Barcelona. His research interests include IR, evaluation of search algorithms, hyperlink structure analysis, structured document retrieval, combination of evidence, distributed IR, as well as efficiency and scalability of search engines. <vassilis@yahoo-inc.com>.

such as index pruning and snippet generation; many automatic query expansion and re-formulation techniques [1] [4] [5]; and a comprehensive set of query performance predictors [3]. Terrier also implements various powerful compression techniques and distributed architectures [2], which make it a suitable retrieval platform for large-scale collections that may be used both in a centralised and a distributed setting. Finally, Terrier implements an effective, highly configurable, and scalable in-house crawler called Labrador[2], which has been used for the deployment of Terrier in various industrial applications.

The remainder of this paper is organised as follows. Section 2 presents the Divergence From Randomness framework of parameter-free probabilistic retrieval models, which is implemented in Terrier. Section 3 introduces Terrier's overall architecture, while Section 4 presents cutting-edge research that is implemented in Terrier. Section 5 summarises Terrier's main features and research areas.

2 Divergence From Randomness Retrieval Models

Search engines return relevant documents to user queries by estimating the relevance of the document content to queries. Document content is estimated using retrieval or weighting models. Terrier implements a variety of such weighting models. One of the highly effective innovations in IR research implemented in Terrier is the Divergence From Randomness (DFR) framework for deriving parameter-free probabilistic weighting models for IR [1]. This section presents an overview of the DFR retrieval models (Section 3.2 describes the integration of retrieval models in Terrier).

The DFR models estimate the informative content of a document using lexical frequency statistics, such as the frequency of a term in a document/collection, the number of documents in which a term appears, and so on. The DFR models are based on a simple idea:

"The more the divergence of the within-document term-frequency from its frequency within the collection, the more the information carried by the term in the document".

The DFR models comprise three components, namely

- a *randomness* model,
- an *information gain* model,
- a *term frequency normalisation* model.

The *randomness* model estimates the probability that, within a document collection, a term occurs in a document randomly. According to the above DFR idea, the less randomly a term occurs in a document, the more information it conveys. Specifically, given a collection D of documents, the *randomness* model RM estimates the probability $P_{RM}(t \in d \mid D)$ of having tf occurrences of a term t in a document d. The importance of term t in document d corresponds to the informative content $-\log_2(P_{RM}(t \in d \mid D))$.

The *information gain* model estimates the probability that, within a document collection, a term is a good descriptor of a document. Specifically, the *information gain* model GM estimates the informative content $1-P_{risk}$ of the probability P_{risk} that a term t is a good descriptor for a document. Good descriptors are terms which have a low frequency in the whole collection, but a high frequency in the subset of documents within the collection that are relevant to the user query. One of the models used in DFR to compute the probability P_{risk} that a term is a good descriptor for a document is the Laplace after-effect model: $P_{risk} = tf / (tf + 1)$, which estimates the probability of having one more occurrence of a term in a document, after having seen it tf times already.

The *term frequency normalisation* model adjusts the frequency of a term in a document, on the basis of the length of that document and the average document length in the whole collection, so that longer documents do not have an unfair advantage over shorter documents. For example, the Normalisation 2 *term frequency normalisation* model assumes a decreasing density function of the normalised term frequency with respect to the document length l, so that the normalised term frequency tfn is:

$tfn = tf \cdot \log_2(1 + c \cdot (avg\_l / l))$, where avg_l is the average document length in the collection, l is the length of a document d, and c is a hyper-parameter.

The final relevance score of a document d for a query q is given by the DFR model as follows:

$$w_{d,q} = \sum_{t \in q} qtw \cdot w_{d,t}$$

where

$$w_{d,t} = (-\log_2 P_{RM}) \cdot (1 - P_{risk})$$

where $w_{d,t}$ is the weight of the term t in document d, $qtw = qtf / qtf_{\max}$, qtf is the frequency of term t in the query q, and qtf_{max} is the maximum qtf in q.

Different *randomness*, *information gain*, and *term frequency normalisation* models can be used to generate many different retrieval models – a feature which renders the DFR family of models flexible and easily extensible. For example, if P_{RM} is estimated using the Poisson randomness model, GM is estimated using the Laplace after-effect model, and tfn is computed according to Normalisation 2 (described above), then the resulting weighting model is denoted by PL2.

Terrier includes a selection of effective retrieval models based on DFR. Furthermore, Terrier also implements various extensions of DFR models, which efficiently integrate several types of evidence, such as query term dependence/proximity and document structure term statistics, into the actual DFR model [4]. The integration of this kind of 'contextual' evidence into the actual matching function that estimates the similarity between a query and a document is a novel and theoretically-elegant feature of the Terrier platform, whose effectiveness has been successfully evaluated [4]. Apart from ranking estimated relevant documents, DFR models can also be used for query expansion [1], which

[2] <http://ir.dcs.gla.ac.uk/labrador/>.

is a retrieval enhancing technique applied using an initial ranking of documents (query expansion is separately presented in Section 4.1). The DFR models are information-theoretic models, for both document ranking and query expansion, a versatile feature that is not possessed by any other IR model. Moreover, the fact that they do not require tuning makes them better-suited for dynamic collections and consequently Web search.

3 Architecture of the Terrier Platform

This section presents the overall retrieval architecture of the Terrier platform. There are two main components in the overall architecture of the Terrier platform, namely

- *indexing* (described in Section 3.1).
- *retrieval* (described in Section 3.2).

Indexing describes the process during which Terrier parses a document collection and represents the information in the collection in the form of an index that contains statistics on term frequency in each document and in the whole collection. *Retrieval* describes the process during which Terrier weights each document term and estimates the likely relevance of a document to a query, on the basis of these term weights. For Web search, Terrier includes a powerful crawler (described in Section 3.4). Moreover, Terrier provides a flexible query language, allowing users to formulate specific preferences in their queries (described in Section 3.3).

3.1 Indexing

This subsection describes the first process in the retrieval architecture of Terrier, during which the document collection is parsed and the information contained in it is appropriately indexed. Terrier achieves modularity in indexing collections of documents by splitting the process into four stages, where, at each stage, plugins can be added to alter the indexing process. For example, Terrier's various document parsers allow it to index HTML documents, plain text documents, Microsoft Word, Excel, PowerPoint documents and Adobe Acrobat (PDF) files. In addition, Terrier allows the direct parsing and indexing of compressed collections. Such collections of documents can be static, or generated dynamically by a Web crawler such as Labrador. Overall, Terrier's modular architecture allows flexibility in the indexing process at several stages:

- in the handling of a collection of documents.
- in handling and parsing each individual document.
- in the processing of terms from documents.
- in writing the index data structures.

During indexing, Terrier assigns to each term extracted from a document three fundamental properties, namely

- the actual string textual form of the term.
- he position of the term in the document.
- the document fields in which the term occurs (fields can be arbitrarily defined by the document plugin, but typically relate to HTML/XML tags).

During indexing, the terms pass through a highly-configurable 'Term Pipeline', which transforms them in various ways, using plugins such as n-gram indexing, stemming, removing stopwords in various languages, expanding acronyms, and so on. The outcome of the Term Pipeline is passed to the Indexer, which writes the four main data structures of the index, namely a Lexicon, an Inverted Index, a Document Index and a Direct Index.

- The **Lexicon** stores global statistics about each term that occurs in the collection, namely the number of times it appears, and the number of different documents it appears in. Moreover, to facilitate retrieval, the entry for each term in the lexicon contains a pointer to the corresponding posting list in the Inverted Index.
- The **Inverted Index** stores the postings list of a term, which is a list of the document identifiers (ids) that the term occurs in, and the frequencies of the term in those documents. Optionally, the postings list can also contain the positions or the fields (e.g. HTML tags) in the document that the term occurs in. Positional information allows phrasal and proximity search to be performed. The document ids are encoded in the Inverted Index using Gamma encoding, the term frequencies are encoded using Unary encoding, and the term positions, if recorded, are encoded using Gamma encoding.
- The **Document Index** stores for each document, the document length and the pointer to the corresponding entry in the Direct Index.
- The **Direct Index** stores for each document, the term ids and term frequencies of the terms in the document. The Direct Index can be used to facilitate easy and efficient query expansion, which is described in Section 4.1, or document clustering and grouping (such as in Vivisimo[3]). Similarly to the Inverted Index, the positions of each term in the document can also be stored in the Direct Index.

The Direct Index contents are compressed in an orthogonal way to the Inverted Index. Term ids are written using Gamma encoding, while term frequencies are written using Unary encoding. Term positions, if recorded, are encoded using Gamma encoding.

The index data structures described above are highly compressed allowing large collections of documents to be efficiently indexed for retrieval using little disk space. Moreover they can easily be extended or replaced with alternative data structures tailored to specific applications.

This subsection has described the first stage in the overall retrieval process, namely indexing, during which the document collection is parsed and the information contained in it is indexed. Terrier's modular architecture consists in splitting the indexing process into four stages, and allowing for any number of plugins to tailor specific indexing needs, in an effective, efficient and highly configurable way.

3.2 Retrieval

This subsection describes the second process in Terrier's overall retrieval architecture, during which documents

[3] <http://search.vivisimo.com/>.

relevant to queries are found and ranked on the basis of this estimated relevance. Document relevance is estimated using weighting models, such as those in the DFR framework (Section 2). Specifically, each query term in a document is assigned a weight, which captures the importance of that term to the document. Term weights are then used to match documents to a query, and rank documents according to their estimated relevance to the query.

Unlike other retrieval platforms, Terrier supports a great range of statistically-different weighting models, so as to facilitate research and cross-comparison of different retrieval strategies. Specifically, the open source 1.0.2 version of Terrier supports eight DFR document weighting models, all of which perform robustly on standard document collections. Additionally, the open source 1.0.2 version of Terrier also includes numerous forms of the classical TF-IDF, Ponte-Croft's Language Modelling 291008 and the well-established Okapi's BM25 probabilistic weighting models.

Finally, Terrier is flexible in allowing the scoring of documents to be altered at various stages in the retrieval process, to take into account additional types of evidence. These stages are score modifiers, post processors and post filters. Section 4 describes various forms of cutting-edge research that takes into account additional types of retrieval evidence, often implemented in Terrier using these facilities.

3.3 Query Language

Terrier includes a powerful query language that allows for user preferences to be taken into account, by letting the user specify additional operations on top of conventional queries. Such operations may specify that a particular query term should or should not appear in the retrieved documents. Other available operations include requiring that terms appear in particular fields, phrase queries or proximity queries. Note that these operations could not have been made possible without Terrier's modular and powerful indexing functionalities. An overview of the available query language operations is given below.

- `t1 t2` retrieves documents with either query term t1 or query term `t2`.
- `t1^ 2.3` sets the weight of query term `t1` to 2.3.
- `+t1 -t2` retrieves documents with query term `t1` but not query term t2.
- `"t1 t2"` retrieves documents where the query terms `t1` and `t2` occur next to each other.
- `"t1 t2"~n` retrieves documents where the query terms `t1, t2` occur within n terms of each other.
- `+(t1 t2)` specifies that both query terms t1 and t2 are required.
- `field:t1` retrieves documents where query term `t1` must appear in the specified field.
- `control:on/off` enables or disables a given control. For example, query expansion is enabled with `qe:on`.

Such user preferences enhance the user-friendliness and usability of the system, while at the same time, allowing for more flexibility and granularity in the actual retrieval process. This query language functionality is applied after the indexing and retrieval stages of Terrier's overall retrieval architecture, which have been described in Sections 3.1 and 3.2, correspondingly.

3.4 Crawler

Terrier includes an effective, highly configurable and scalable in-house crawler called Labrador. Labrador is a distributed Web crawler (or spider), written in Perl, and fully integrated with the Terrier Web search framework. Labrador is suitable for crawling a single website or intranet, up to large-scale Web/Internet crawls. A modular interface allows customisation of filtering and crawler strategies to suit the application.

Labrador operates in a distributed architecture, whereby all crawler processes are controlled by a manager, known as the dispatcher. New Web sites to be crawled are allocated to crawler processes, whereafter each crawler handles all URLs for a given site. Inter-site links are passed back to the dispatcher, to be passed on to the appropriate crawler, or for a new site, allocated a crawler. A typical deployment of the Labrador Web crawler is shown in Figure 1.

4 Cutting-Edge Research in Terrier

Terrier addresses a variety of cutting-edge research areas, aiming to enhance retrieval performance in all of these areas. The remainder of this section presents some parts of this research.

4.1 Query Expansion

Search engines use query expansion to retrieve relevant documents which may not contain any occurrences of the

Web search task (TREC)	Uniform	Selective
Home page finding (2003)	0.6498	0.7658
Home page finding (2004)	0.5555	0.7025
Named page finding (2003)	0.6836	0.7827
Named page finding (2004)	0.6814	0.8019

Table 1: Mean Average Precision of Retrieval with Field-based Weighting Models Applied uniformly versus selectively per Query.

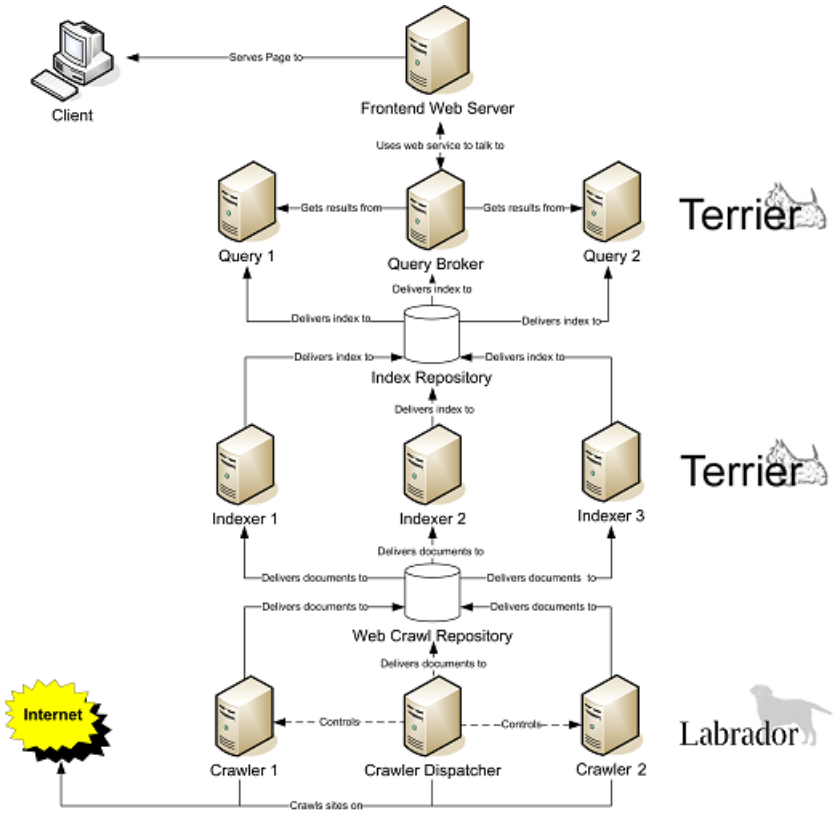

Figure 1: Retrieval Architecture of Terrier Deployed in a Web Search Setting.

user's query terms. In query expansion, the query is enriched with more relevant terms, in order to facilitate the retrieval of additional relevant documents. Terrier implements a state-of-the-art automatic query expansion, which enhances retrieval performance, by taking the top most informative terms from the top-ranked documents of the query, and adding these new related terms to the query. This operation is made possible by the presence of the Direct Index, which allows the terms and their frequencies to be determined for each document in the index (see Section 3.1). The expanded query is re-weighted and rerun, providing a richer and better set of retrieved documents.

Even though automatic query expansion is a highly effective mechanism for many IR tasks, the decision of applying query expansion may be affected by

(1) the type and size of collection used, and

(2) the difficulty of queries.

Terrier addresses point (1) by implementing an automatic tuning of the inherent query expansion parameters, and point (2) by implementing several tools for determining possible indicators that predict the query difficulty, and hence determine the potential applicability of query expansion [3]. Terrier's default query expansion mechanism is a highly effective model from the DFR family called Bo1 [1] [7], which gives effective retrieval performance, even with only using 3 top-ranked retrieved documents and a handful of additional query terms. Terrier also provides several term weighting models from the DFR framework which are useful for identifying informative terms from top-ranked documents, as well as some well-established query expansion techniques, such as Rocchio's method. Terrier's query expansion mechanism is also available in interactive mode, a feature that is well suited for Web search.

4.2 Syntactically-based Information Processing

Terrier implements a novel low-cost natural language processing (NLP) model, which automatically identifies informative content in text. This information processing model is a statistical model that has several applications, ranging from index pruning, to query reformulation, and snippet (summary) generation. The NLP model uses part-of-speech patterns to automatically identify the presence and absence of informative content in text [5]. Since this model does not make use of deep grammatical formalisms, it is neither computationally costly, nor resource-demanding.

Terrier implements three applications of this NLP model, namely (i) query reformulation, (ii) index pruning, and (iii) snippet generation.

Syntactically-based Query Reformulation: This technique reformulates verbose queries, so that they include less estimated noise, hence increasing their informativeness and the likelihood of them fetching more relevant documents [5]. The reduction of noise is realised using solely syntactically-

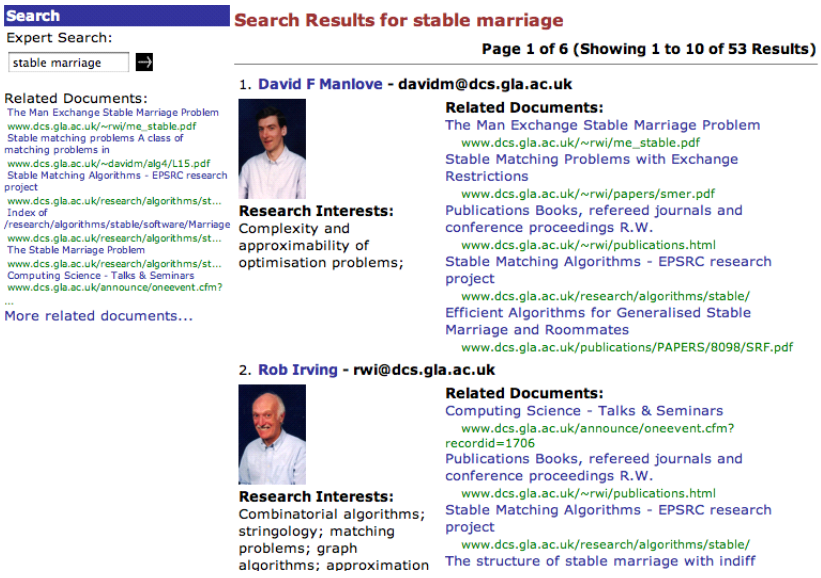

Figure 2: Example Search Results from a Terrier Expert Search System in an Enterprise Setting.

based query-/document-independent evidence, a feature which is very original and low-cost. Syntactically-based query reformulation can be used in domain-specific retrieval applications where users formulate verbose queries, such as the digital libraries domain, or applied after phrase-based query expansion to make sure that no noisy fragments have been accidentally added to the query.

Syntactically-based Index Pruning: This technique prunes estimated noise from all the data structures of the index [4]. The aim is to improve system efficiency, by producing an index that is more economical to store and to query, at no detrimental cost to retrieval performance.

Syntactically-based Snippet Generation: This technique automatically generates estimated informative snippets of documents. Such snippets can be used either (a) in system-internal processes, such as snippet-assisted query expansion for example, or (b) offered to the users to facilitate their understanding of the returned documents. For example, these could be the snippets of the relevant returned pages offered by Web search engines to the users.

4.3 Web Search

Terrier implements certain functionalities that address specific characteristics of retrieving information from the Web. Specifically, in a large-scale setting involving millions of documents, Terrier makes use of distributed indexing and querying across multiple servers. The efficiency of this architecture has been the subject of careful research using simulations and real-world implementations [2]. An overview of a typical Web search operation using Terrier is displayed in Figure 1. Terrier uses information not only from the content of Web documents, but also from features such as the way in which documents are linked to one another on the Web, the URL path/structure and anchor text information of Web documents, and so on. This section presents some of these features of the Web documents, and the specific research implemented in Terrier that takes them into account in order to enhance retrieval performance.

Firstly, the HTML tags of Web documents and the text of a document's URL address can be indicative of the content of a document. Moreover, the anchor text of the incoming hyperlinks to a document can serve as very good descriptors of the document's content. The above features constitute field evidence, which Terrier uses to enhance retrieval performance. As presented in Section 2, Terrier implements several field-based weighting models which take into account these different sources of textual evidence in a fine-grained manner, in order to produce an accurate ranking of relevant documents [4] [9] [11].

Secondly, the hyperlink patterns of how Web pages are linked to each other can indicate the likely relevance of a Web page to a query. For example, the authority of Web documents can be estimated by examining the link graph of the Web.

Terrier takes advantage of this information by using the Absorbing Model [14] link popularity scores to improve retrieval performance on Web tasks [4] [7] [11] [12] (PageRank and other link analysis techniques are also supported). Terrier also utilises the fact that Web documents often have hyperlinks to other similar documents and may form thematic clusters, in order to enhance the retrieval of relevant documents [11].

Moreover, not all Web queries may benefit equally from applying the same retrieval approach. Terrier implements a statistical decision mechanism that selects an appropriate retrieval approach on a per-query basis [11] [13] [15]. The selection of a particular retrieval approach is based on the outcome of a low-cost statistical predictor, which is performed before the final ranking of the retrieved documents. The predictor is a process that extracts features from a sam-

ple of the set of retrieved documents. Example predictors might:

- examine the count of occurrences of query terms in retrieved documents, which indicates the extent to which the query is covered in the document collection.
- consider information from the distribution of retrieved documents in larger aggregates of related Web documents, such as whole Web sites or directories within a Web site.
- estimate the usefulness of the hyperlink structures among a sample of the set of retrieved Web documents.

Table 1 displays the improvement in retrieval performance marked when field-based weighting models are used selectively on a per-query basis, as opposed to for all queries (from [11], Table 6.1, page 134). Retrieval performance is measured using Mean Average Precision (MAP), on Web retrieval tasks from the Text REtrieval Conference (TREC) 2003 and 2004. A thorough evaluation of the selective application of various Web IR techniques in combination with predictors such as those above is covered in [11].

Terrier supports Web search across a selection of languages, in both mono-lingual and multi-lingual settings, and has been successfully tested on eleven languages [9].

4.4 Corporate Web Search

With the advent of "knowledge workers" in large collaborative enterprise organisations, there has been an explosion in the number of intranets (small internal corporate Webs). The retrieval technologies implemented in Terrier scale very well to such smaller corpora, making Terrier an effective tool to increase the productivity of knowledge workers.

Moreover, it has been found that corporate users often have the need not to find relevant documents in their intranet, but also to identify people with relevant expertise. An Expert Search system aids users with their expertise need, by identifying people with relevant expertise. Terrier addresses the expert search problem by implementing a novel and very effective retrieval model, namely the Voting Model for Expert Search [8]. At indexing time, for each candidate expert, a profile of textual expertise evidence is accumulated - for example, the documents authored by each candidate. Instead of directly ranking candidates with respect to the query, the Voting Model considers the ranking of documents with respect to the query. Then the number of votes each candidate receives from their associated profile documents in the document ranking indicates how much expertise the candidate has in the required topic area. The votes for each candidate expert are then appropriately aggregated to form a ranking of candidate experts, taking into account the number of voting documents for that candidate, and the relevance score of the voting documents. The Voting Model is extensible and general. For example, Terrier implements eleven techniques for aggregating votes for candidates, some of which have been further extended, so as to account for candidate profile length [4]. This candidate profile length normalisation, which is an adaptation of the term frequency normalisation model from the DFR framework (Section 2), is used to control any bias towards candidates with longer profile lengths. Figure 2 presents example search results from a Terrier expert search system in an Enterprise setting. Overall, Terrier implements a state-of-the-art retrieval model for expert search, which ranks candidates according to the extent to which their associated documents are retrieved by the underlying document ranking on the topic.

While the usefulness of expert search can easily be seen within the Enterprise setting, research in this setting is equally applicable in the blogosphere. For blog search engines, users are often looking to find new bloggers to read that have a recurring interest in a topic, and for this scenario, existing research into expert search will be very applicable.

5 Conclusions

As the map of information flow changes dynamically, by expanding in size, content and means of exchange, new trends emerge. The research that is put into Terrier addresses these trends, by identifying and modelling ways, which allow for information to be effectively and efficiently retrieved, especially from heterogeneous environments such as the Web. Terrier is a high performance, flexible, robust, transparent and scalable search engine [10], which implements cutting-edge ideas from probability and information theory, statistics, natural language processing, and data compression techniques, to name but a few. Examples of such cutting-edge research include: the highly effective Divergence From Randomness (DFR) family of Information Retrieval (IR) models [1]; several information analysis and selective combination of evidence methods tailored to Web Search [4] [7] [11] [13] [14] [15]; a novel syntactically-based information processing model [4] [6], with applications such as index pruning and snippet generation; many automatic query expansion and re-formulation techniques [1] [5]; and a comprehensive set of query performance predictors [3]. Though driven by Web search initially, Terrier has been extended to Desktop and Web corporate search and blogosphere applications. The flexible, extensible and highly configurable architecture of Terrier allows for all of the above functionalities to be implemented into the system efficiently and effectively.

References

[1] G. Amati. Probabilistic Models for Information Retrieval based on Divergence from Randomness. PhD thesis, Department of Computing Science, University of Glasgow, 2003.

[2] F. Cacheda, V. Carneiro, V. Plachouras, and I. Ounis. Performance analysis of distributed information retrieval architectures using an improved network simulation model. Information Processing & Management, Elsevier, 2007.

[3] B. He, and I. Ounis. Query Performance Prediction. In Information Systems, Elsevier, 2006.

[4] C. Lioma, C. Macdonald, V. Plachouras, J. Peng,

B. He, and I. Ounis. University of Glasgow at TREC2006: Experiments in Terabyte and Enterprise Tracks with Terrier. In TREC '06: Proceedings of the 15th Text REtrieval Conference (TREC 2006), November, 2006, NIST.

[5] C. Lioma, and I. Ounis. A Syntactically-based Query Reformulation Technique for Information Retrieval. In Information Processing and Management, Elsevier, 2007.

[6] C. Lioma and I. Ounis. Examining the Content Load of Part-of-Speech Blocks for Information Retrieval. In Proceedings of COLING/ACL 2006, Sydney, Australia, 2006.

[7] C. Macdonald, B. He, V. Plachouras, and I. Ounis. University of Glasgow at TREC2005: Experiments in Terabyte and Enterprise Tracks with Terrier. In TREC '05: Proceedings of the 14th Text REtrieval Conference (TREC 2005), November, 2005, NIST.

[8] C. Macdonald and I. Ounis Voting for Candidates: Adapting Data Fusion Techniques for an Expert Search Task In Proceedings of the CIKM'2006 , Arlington, VA. 2006.

[9] C. Macdonald, V. Plachouras, H. Ben, C. Lioma, and I. Ounis. University of Glasgow at WebCLEF 2005: Experiments in per-field normalisation and language specific stemming. C. Peters, F. Gey, J. Gonzalo, H. Mueller, G. Jones, M. Kluck, B. Magnini and M. de Rijke, editors, CLEF 2005, volume 4022 of Lecture Notes in Computer Science, pages 898–907. Springer, 2006.

[10] I. Ounis, G. Amati, Plachouras V., B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006), Seattle, USA, 2006.

[11] V. Plachouras. Selective Web Information Retrieval. PhD thesis, Department of Computing Science, University of Glasgow, 2006.

[12] V. Plachouras, B. He, and I. Ounis. University of Glasgow at TREC2004: Experiments in Web, Robust and Terabyte tracks with Terrier. In Proceedings of the TREC-2004, Gaithersburg, MD. 2004.

[13] V. Plachouras, and I. Ounis. Usefulness of hyperlink structure for query-biased topic distillation. In SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and Development in information retrieval, Sheffield, UK, 2004.

[14] V. Plachouras, I. Ounis, and G. Amati. The Static Absorbing Model for the Web. Journal of Web Engineering, 4(2):165–186, 2005.

[15] V. Plachouras, F. Cacheda, and I. Ounis. A Decision Mechanism for the Selective Combination of Evidence in Topic Distillation. Information Retrieval, 9(2):139–163, 2006.

[16] J. M. Ponte and W. B. Croft. A language modelling approach to information retrieval. In SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pages 275–281, Melbourne, Australia, 1998.

