UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING, SCIENCE & MATHEMATICS

SCHOOL OF ELECTRONICS & COMPUTER SCIENCE

SEMIOMETRICS: PRODUCING A COMPOSITIONAL VIEW OF INFLUENCE

By

Duncan M. McRae-Spencer

A thesis submitted for the degree of Doctor of Philosophy

March 2007

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE & MATHEMATICS

SCHOOL OF ELECTRONICS & COMPUTER SCIENCE

Doctor of Philosophy

SEMIOMETRICS: PRODUCING A COMPOSITIONAL VIEW OF INFLUENCE

by Duncan M. McRae-Spencer

High-impact academic papers are not necessarily the most cited. For example, Einstein's 'Special Relativity' paper from 1905 received (and continues to receive) fewer citations from other papers than his 'Brownian Motion' paper of the same year, despite the former radically changing the course of an entire scientific discipline to a much greater extent. Similarly, 'impact' metrics using citation count alone are, it is argued, not adequate for determining the scientific influence of papers, authors or small groups of authors. Although valid, they remain controversial when used to determine influence of larger groups or journals. While the term 'impact' has become closely linked to a journal's citation-based Journal Impact Factor score, this thesis uses the term 'influence' to describe the wider effectiveness of research, combining citation and metadata analysis to allow richer calculations to be performed over large-scale document networks. As a result, more qualitative influence ratings can be determined and a broader outlook on scientific disciplines can be produced. These ratings are best applied using an ontology-based data source, allowing more efficient inference than under a traditional RDBMS system, and allowing easier integration between heterogeneous data sources. These metrics, termed 'Semantic Bibliometrics' or 'Semiometrics', can be applied at a variety of levels of granularity, allowing a compositional framework for impact and influence analysis. This thesis describes the process of data preparation, systems architecture, metric value and data integration for such a system, introducing novel approaches at all four stages, thereby creating a working semiometrics system for determining influence at different semantic levels of granularity.

Contents

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Scope and Central Hypothesis 2

1.2.1 Scope 2

1.2.2 Central Hypothesis 2

1.3 Thesis Structure 4

1.4 Contributions 5

Chapter 2 Literature Review 7

2.1 Scientific Publishing and Citation Analysis 7

2.1.1 The ISI Journal Impact Factor 8

2.1.2 Criticisms of Citation Counting and the Impact Factor 10

2.1.3 Relative citation scores might be an improvement 14

2.1.4 Other possible measures: beyond bibliometrics 15

2.2 Large-scale Digital Libraries 20

2.2.1 Measuring Citations 22

2.2.2 Move to Open Archiving 23

2.3 The Semantic Web 25

2.3.1 Application to Citation/Digital Library culture 27

2.3.2 Use of ontologies in academic paper research 28

2.4 Knowledge Reuse and Mapping Knowledge Domains 28

2.4.1 Coauthorship networks and patterns of scientific collaboration. 29

2.4.2 Characterizing PNAS: Impact Maps 29

2.4.3 Mapping topics and topic bursts in PNAS 30

2.4.4 Visualizing a knowledge domain with cartographic means 30

2.5 Metadata Transformation and Preparation 31

2.5.1 Identity Uncertainty 32

2.5.2 Data Manipulation Techniques 33

Chapter 3 Author Disambiguation 35

3.1 The AKTiveAuthor Problem Domain 36

3.2 Overview of the AKTiveAuthor system 36

3.3 Methodology and experiments 38

3.3.1 Step one: Initial clustering 39
3.3.2 Step two: Self-citation analysis 40
3.3.3 Step three: Co-author analysis 40
3.3.4 Step four: Source URL analysis 40
3.3.5 Step five: Sanity name-check 41
3.4 Metrics 41
3.4.1 D-Precision 42
3.4.2 D-Recall 42
3.4.3 D-F-measure 43
3.5 Results 43
3.5.1 D-precision higher than d-recall 44
3.5.2 All three methods are required for good results 45
3.5.3 Certain types of author are better suited to these analyses 46
3.6 Conclusions 47
Chapter 4 Ontology-Assisted Data Mediation 49
4.1 Problem Domain 50
4.2 Overview of solution 50
4.3 Motivation 51
4.4 Data Storage Models and Purpose 53
4.5 Experiment Details 54
4.5.1 RDBMS approach 55
4.5.2 RDF/Ontology approach 57
4.5.3 Client 1: Ordered Lists and Summaries 58
4.5.4 Client 2: Semiometrics Viewer 59
4.5.5 Experiments 59
4.6 Results 60
4.6.1 Summary of results 60
4.6.2 Discussion of results 61
4.6.3 Further Analysis 63
4.7 Conclusion 64
Chapter 5 Practical Semiometrics 66
5.1 Problem Domain 66
5.2 Overview of the Semiometrics System 67
5.3 Empirical Evaluation 67

5.4 The Semiometrics Approach 68
5.4.1 Categorisation 68
5.4.2 Metrics 69
5.4.3 Avoided Metrics 73
5.4.4 Interface 75
5.5 Experiment 76
5.5.1 Limitations and Expectations 77
5.5.2 Data and details 78
5.5.3 Results 79
5.6 Comparisons 94
5.7 Conclusions 95
Chapter 6 Metadata Integration 97
6.1 Problem Domain 97
6.2 Overview of system 98
6.3 Empirical Evaluation 99
6.3.1 Chapter 4 Revisited 100
6.3.2 In-depth View of Matching Process 100
6.3.3 Paper Matching 101
6.3.4 Author Matching 103
6.3.5 Prototyping: three potential match-points 105
6.3.6 Metrics 108
6.4 Results and discussion 109
6.4.1 Paper Matching 109
6.4.2 Author Matching 111
6.4.3 Usefulness of data 113
6.4.4 Programmatic Changes 113
6.5 Conclusion 114
Chapter 7 Conclusion 115
7.1 Conclusions 115
7.2 Future direction of work 118
7.2.1 Immediate improvements to process and applications 118
7.2.2 Ontology Usage 119
7.2.3 Incorporate with OAI:PMH 120
7.2.4 More measures, more levels 120

7.3 The future and Semantic Digital Libraries 120

Appendix A **122**

Appendix B **130**

References **137**

Index **147**

List of Figures

Figure 2.1 : Chen's Citespace showing discipline development over time.............. 16
Figure 2.2 : Citeseer's citation count graph for a typical paper............................... 23
Figure 2.3 : Citebase statistics for a typical paper.. 25
Figure 2.4 : Skupin's geographical representation of a term-dominance landscape 31
Figure 3.1: Partial citation graph for Nick Jennings 37
Figure 3.2 : Results for the eight name-clusters... 44
Figure 4.1 : The database schema for the MySQL sources for the ACM dataset.... 55
Figure 4.2 : Screenshot showing the semantic web services client pages................ 57
Figure 4.3 : Screenshot of SemioViewer application... 58
Figure 4.4 : Results of experiments performed on ACM dataset............................. 61
Figure 5.1 : The AKT Semiometrics Website... 75
Figure 5.2 : Semiometrics Viewer Applet.. 76
Figure 5.3 : List-edit distance results for papers 84
Figure 5.4 : List-edit distance results for authors... 90
Figure 6.1 : Overview of system as described in previous chapters 98
Figure B.1 : AKTiveAuthor Implementation Details... 130
Figure B.2 : Semiometrics system implementation details 132
Figure B.3 : System amendments to allow for semantic data integration.............. 134

Declaration

This thesis is based upon work undertaken by the author. What is presented is the original work of the author with the exception of existing toolsets used as part of the processes, which are credited in text as appropriate. This work has been supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. Parts of this thesis have appeared in the publications [McRae-Spencer & Shadbolt 2006] and [McRae-Spencer & Shadbolt 2006a], and these parts are marked in the text as appropriate.

Acknowledgements

I would firstly like to thank my supervisor Nigel Shadbolt for his wise advice, careful steering and useful connections, not necessarily in that order. Learning how to research and produce publications is much easier with a supervisor who has as much experience and enthusiasm as Nigel.

I would also like to thank the Southampton AKT team for help, collaboration and support, in particular Harith Alani, Steve Harris, Nick Gibbins, Hugh Glaser and also Susan Davies, who makes it all work.

Wider thanks are also due to a members of Southampton's IAM research group for their encouragement and often crucial insights, in particular Les Carr, Tim Brody, Nick Jennings, Wendy Hall and Srinandan Dasmahapatra. Other individual and groups also deserve thanks for ideas, collaboration, advice and even toolsets, particularly Sam Chapman and the NLP group at Sheffield University and Mounia Lalmas and Theodora Tsikrika from the Information Retrieval group of Queen Mary, University of London.

For the Citeseer sections of this work I am indebted to Isaac Councill and C. Lee Giles of Penn State University for their provision of Citeseer and ongoing useful comments and suggestions.

The ACM dataset was acquired for ECS by Wendy Hall and provided to me by Vladimir Mircevski. ACM papers provided by Stephen Ball. Thanks are due to them all for this invaluable data source.

This work has been supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01.

Definitions and Abbreviations Used

Bibliometrics – The study, or measurement, of texts and information. Within the academic domain this often refers to tracing citation patterns between papers.

Bubble-sort – A simple, inefficient sorting algorithm where list items are compared with their list neighbours, swapping places if appropriate.

Citation – A reference from one item to another. Typically with academic papers, citations are placed in a bibliography at the end of the paper.

Digital Library – Central store of papers stored in digital format, typically searchable and available via a web portal.

Disambiguation – The process by which similar and potentially conflicting ("ambiguous") data are separated or merged as necessary.

Dublin Core – A widely-adopted set of standard metadata fields (fifteen at the most basic level) that can be applied across a range of documents.

Graph – A set of items ("nodes" or "vertices") connected by edges.

Half-Life – A term reflecting the time required for half the nuclei in a sample of a specific isotope to undergo radioactive decay. The graphical curve produced by measuring such a phenomena is mirrored by a graph representing the number of citations a document receives over time, as discussed more fully in section 2.1.4.

Impact / Impact Factor – The scientific impact of a paper. In the academic domain this term has become closely associated with the specific measure that is the 'ISI Journal Impact Factor'; for this reason, the word 'influence' has tended to be used in this thesis rather than 'impact' in this context.

Influence – A term broader than impact that has been used in this thesis to represent scientific effectiveness according to a wide variety of measurements.

JCR – Journal Citation Report, published by ISI, detailing Journal Impact Factor scores.

JIF – Journal Impact Factor.

Metadata – Data about data. Typically, for a document, metadata includes information such as author name, date of creation and document title.

Ontology - Formal definitions of relationships among terms, typically within a specific domain.

OWL – The Web Ontology Language, used to express ontologies in a manner compatible with XML/RDF.

RDBMS – Relational DataBase Management System, a type of data storage facility capable of storing data across relationally-linked tables.

RDF – The language of the Semantic Web, based on URI identification and subject-predicate-object triple relationships.

Semantic Web – A future iteration of the World Wide Web with more structured, machine-readable data.

SPARQL - SPARQL Protocol And RDF Query Language, a standard syntax for querying triplestores.

Triplestore – Data repository designed for storage and retrieval of RDF triples.

W3C – The World Wide Web Consortium, the main international standards organisation for the web.

XML – Extensible Markup Language, a W3C-standard general purpose markup language. RDF and OWL are formally-defined languages based on XML.

Chapter 1
Introduction

1.1 Motivation

High-impact academic papers are not necessarily the most cited. For example, Einstein's 'Special Relativity' paper from 1905 [Einstein 1905] received (and continues to receive) fewer citations from other papers than his 'Brownian Motion' paper of the same year [Einstein 1905a], despite the former radically changing the course of an entire scientific discipline to a much greater extent. Similarly, 'impact' measurements using document and/or citation count alone are, it is argued (eg [Seglen 1997]), not adequate for determining the scientific influence of papers, authors or groups of less than a hundred authors [Opthof 1997]. Although valid, they remain "controversial" [Garfield 1996] when used to determine influence of larger groups or journals.

It is important to specify what is meant by the terms 'impact' and 'influence'. Since ISI began publishing citation scores for journals (the 'Journal Impact Factor') in 1975, the term 'impact' has become closely linked with the JIF scores when considering research quality. As a result of this, this thesis will use the term 'influence' to refer to a broader set of metrics that can be applied to research publications and researchers. While 'impact' has come to mean a journal's score, the aim here is to use the term 'influence' to describe the wider effectiveness of research, combining citation and metadata analysis to allow richer calculations to be performed over large-scale document networks. As a result, more qualitative influence ratings can be determined and a broader outlook on scientific disciplines can be produced.

This thesis examines issues associated with research influence measurements. The thesis explores the idea of using an ontology to represent academic paper metadata, including citations, authors, topics and affiliations, and empirically evaluates some of the benefits associated with such an approach. The evaluations presented are grounded in the domain of calculating and composing influence scores in the field of computer science research.

1.2 Scope and Central Hypothesis

1.2.1 Scope

The calculation of the kind of influence metrics we are describing requires two main areas of study: firstly the creation of a system capable of being both updated and queried in real time, and secondly the actual metrics to be used.

The scope of this thesis therefore covers questions of metadata storage and retrieval at various levels of granularity (paper, author, institution, venue, discipline etc) along with a discussion of relevant metrics for determining research influence.

The nature of article-centric metadata means that frequently, for many metadata forms (including standard Dublin Core), the paper is the only first-class object in the system. Therefore issues of disambiguation of other data, particularly authors of papers, is central to the creation of a useful real-world system, and this need is therefore also within the scope of this thesis.

1.2.2 Central Hypothesis

The central hypothesis of this thesis is that research influence ratings may be based on semantic bibliometric calculations and performed on graph-disambiguated ontology-represented data. Such ratings offer advantages over traditional influence scores. Four sub-hypotheses are examined in order to test the central hypothesis.

The first sub-hypothesis is that approaches based on citation references can be applied to the problem of document author disambiguation to produce better results than other techniques from the literature, producing a better core of data on which to build a semantic bibliometric system. Specifically, the problem of author disambiguation can be tackled using citation graphs to match document authors and this, combined with other techniques from the literature, produces successful matching to a higher level of effectiveness than other existing techniques such as string-distance matching and machine learning techniques.

The second sub-hypothesis is that using ontologies to represent document metadata allows more efficient traversability and viewing of the different levels (as described in the third sub-hypothesis) than traditional RDBMS approaches. Specifically, the query and update efficiency when using an ontology-based data store for importing and retrieval is better than when performing the same data browsing in a traditional SQL database.

The third sub-hypothesis is that an ontology-based system producing results based on weighted citation scores, graph node authority other such scores reflect more accurately the relative influence of a paper than citation counting alone, and that given the ontological approach, these scores can be amalgamated and viewed at sub-discipline levels to give relative influence scores for documents and authors. Specifically, expert opinion in particular subject domains consistently ranks Semiometric results at least comparably with existing metrics, and the overall capabilities of the system significantly out-perform other metric systems such that the described system represents a real, practical advancement in metrics applications.

The fourth sub-hypothesis is that using ontologies allows easier integration of data from other sources, allowing influence ratings to be drawn using a far broader data scope than a single repository, which tend to favour particular individuals. Specifically, the successful amalgamating of two distinct large-scale metadata sources is easier using an ontology than including in a traditional database.

If the first sub-hypothesis is true, then graph-based approaches are a better method of data disambiguation and enrichment than other methods in the literature. If the second sub-hypothesis is true, then ontology-based semiometric influence scores are a better method of metadata storage and retrieval for large-scale citation databases than traditional RDBMS methods. If the third and fourth sub-hypotheses are true, then a useful ontological framework can be created to allow efficient data integration and access to these semiometric influence scores in a straightforward, practical application. If all four sub-hypotheses are true, then a system that integrates and calculates semiometric influence scores from raw document metadata is both feasible and valid as a means of producing a compositional view of influence at a variety of granularities, thus proving the central hypothesis. The objective of the experiments within this thesis are to prove the four sub-hypotheses, and hence the central hypothesis.

1.3 Thesis Structure

The overall structure of the thesis is as follows. It begins by introducing digital libraries, citation-indexed metadata stores and their associated technologies, setting the context for the work described in this thesis. Four experiments are then discussed, with the aim of providing supporting evidence to one of the four sub-hypotheses. The evidence is then collated and summarised, clarifying the support for the sub-hypotheses. This in turn proves the central hypothesis of this thesis.

Chapter Two, *Literature Review*, reviews existing techniques in publication and citation analysis, digital libraries and Semantic Web technologies, along with metadata preparation and transformation techniques to be used in the rest of the thesis.

Chapter Three, *Author Disambiguation*, introduces the specific data preparation issue of author disambiguation, and introduces a novel technique for matching and distinguishing between authors with similar names. The technique is compared to others in the literature and is applied to a real-world dataset, allowing metadata to be prepared for use in experiments in the remainder of the thesis. The conclusions drawn provide evidence to support the first sub-hypothesis.

Chapter Four, *Ontology-Assisted Data Mediation*, introduces the use of ontology-mediated metadata and semantic web technologies as a means of storage and retrieval. This approach is described in depth and compared with traditional RDBMS/SQL techniques, contrasting in terms of both speed of response and flexibility of data. The conclusions drawn provide evidence to support the second sub-hypothesis.

Chapter Five, *Practical Semiometrics*, introduces Semiometric web services and two client applications. Building on the data and structure described in the previous chapters, this system is described in depth and the metrics offered by the system are detailed. An evaluation is performed to measure the effectiveness of the metrics output by the system and the results are compared with other approaches in the literature. The conclusions drawn provide evidence to support the third sub-hypothesis.

Chapter Six, *Metadata Integration*, discusses the further issue of merging multiple sets of metadata. Building on the ontological framework described in chapter four, the issue of paper and author disambiguation is explored, contrasting with alternative possible approaches. An empirical analysis involving merging of two distinct large-scale datasets is performed and the conclusions drawn provide evidence to support the fourth sub-hypothesis.

Chapter Seven, *Conclusions and Future Work*, concludes this thesis by collating the evidence for each sub-hypothesis, and examining how the four sub-hypotheses prove the central hypothesis. The future direction for the work detailed within this thesis is then discussed.

1.4 Contributions

This thesis provides several novel contributions.

The Semiometric web services and client viewer systems use a novel, ontology-based approach to metadata storage and retrieval. It is novel since most digital libraries and document management systems are built upon either standard relational or bespoke table-based/hash-based databases. While ontology-based metadata has been used in research browsing tools, the system here performs calculations to infer new knowledge about research and researchers that cannot be observed purely through browsing. Ontological storage also allows traversing of the metadata graph to compose views of data at different levels of granularity and perform these calculations accordingly, as well as straightforward data integration for data from heterogeneous data sources.

The metrics produced by the system are a variety of novel and previously-suggested metrics. The means and ability to produce the metrics in a compositional manner, drawing data from heterogeneous sources and performing inference calculations, is the novel aspect of the semiometric approach, along with the novel metrics proposed. The presentation methodology of drawing weighted metrics together to produce a compositional view of influence is also novel.

Finally, the citation graph approach to author disambiguation is a novel technique applied to the heavily-studied name ambiguity issue within data retrieval.

Chapter 2

Literature Review

2.1 Scientific Publishing and Citation Analysis

The traditional means for a scientist to announce and disseminate novel research results is the publishing of those results along with the scientist's conclusion, either as a book in its own right (such as Darwin's 'On The Origin Of Species', published in 1859 [Darwin 1859]) or more usually as a paper in a journal or conference (such as Einstein's Special Relativity paper "On The Electrodynamics Of Moving Bodies", published in the German periodical Annalen der Physik in 1905 [Einstein 1905]). This allows other scientists to repeat and later build upon these works, and publish their own results, citing the contribution made by the earlier work. As such, the scientist who performed the novel research retains the respect for their work, but science as a whole is allowed to use it and thus move forward.

However, it is immediately clear that some research is more important than others. A work such as Einstein's Special Relativity paper opened up a whole new field of physics. By contrast his 1949 paper "Why Socialism" [Einstein 1949] had less influence and addressed a different domain. It would have been – and indeed still is – very important for physicists, commentators, academics and others to be aware of the 1905 paper, while it was less important (although no doubt very interesting) for political theorists to be aware of the 1949 paper. This 'relative importance' can be demonstrated empirically by studying citation patterns: these show that the 1905 paper is the second-most cited paper in physics and physical-chemistry since 1945 (his 1905 paper on Brownian motion [Einstein 1905a] receives even more citations [Chalmers

2005], while the 1949 paper led to Einstein being "virtually neglected and ostracised in his later years" [Myers 2003] and the FBI putting together a dossier on him because of his political views. While the 1949 paper is historically and biographically interesting, it is clear that the 1905 paper has greater relative scientific importance, and has had greater influence over the years.

The question therefore becomes: is it possible to quantify this 'relative scientific importance' of papers? The work presented in a published paper must already have passed two tests of importance: firstly it has been written up by the author(s) of the paper, who must consider it a worthwhile exercise, and secondly to be published in a journal or conference proceedings the paper must normally go through a peer review process. These two tests show that the originators and reviewers consider the conclusions of the paper to be worthwhile: however, that is different from 'influence' as discussed above, which deals with how people are using and building on the work presented in the paper.

While journal subscriptions and paper download statistics can begin to show how many people are reading the article, the clearest indicator of scientific usage is the citation: bibliographic references to other papers shows that the papers being referenced had some bearing, or influence, on the paper now being written. The more citations received, the greater the influence of the original paper.

2.1.1 The ISI Journal Impact Factor

To quantify this impact measure further, and after developing techniques for a number of years[Garfield 1955], the Institute for Scientific Information (ISI) began measuring journal 'Impact Factors' [Garfield 1972] in-house in the 1960s and began publishing them in 1975. Based on citation counting of journal papers, the Impact Factor measures the number of citations papers in a given journal receive from other journals during the three years after it is published, and rate the impact of that journal accordingly. It is important to note that only refereed full papers and notes are included in the JIF, and not editorials or letters contained within the journal.

The calculation for journal impact factor can be calculated as follows (based on [Garfield 1994]):

```
A = total cites in 1992
B = 1992 cites to articles published in 1990-91 (subset of A)
C = number of articles published in 1990-91
D = B/C = 1992 impact factor
```

Initially impact factors were introduced “primarily as a bibliographic research tool for retrieval of overlapping research for the benefit of scientists who worked in relative isolation to contact colleagues with comparable interests” [Opthof 1997]. However, since these were first published in 1972, the JIF figures have become the *de facto* standard metric for determining journal impact, and by inference research influence, across sciences and social sciences. Indeed, in terms of research evaluation, journal impact factors are “probably the most used indicator besides a straightforward count of publications.”[Seglen 1997] A more complete review of citation analysis and the JIF can be found in [Moed 2005] and [Wolfram 2003].

In addition to journal Impact Factors, pure citation counting for an individual paper is also in widespread use, although it is harder to ensure all citations are counted: the count should ideally include citations from highly diverse journals across different disciplines, not all of which are taken into account by ISI, along with conference proceedings, books and perhaps online published articles. However, despite this possible lack of coverage, it has been shown that paper (and person) ratings based on citation counting closely mirrors another influence measure, the Research Assessment Exercise [Harnad *et al.* 2003] [Smith & Eysenck 2002]. Despite the nature of the RAE as being an extremely complex social exercise rather than a simple metric, these reports show a correlation between institutional quality and paper citations.

Similarly, the scientific literature digital library Citeseer [Lawrence *et al.* 1999], containing an automated citation indexing system, produces statistics on the most-cited papers and authors. Although there are clear potential shortcomings of these statistics – lack of completion of the citation counts, data sources skewed in favour of computer science and particular sub-disciplines within – the statistics produced are regularly viewed and downloaded.

However, as has been hinted at in this section, there are numerous criticisms that can be levelled at the statistical significance of citation counting and its associated measures. The following section details some of these criticisms and argues that a re-think of pure citation counting as the absolute measure of influence is required.

2.1.2 Criticisms of Citation Counting and the Impact Factor

In investigating the science of citation analysis, it is worth firstly asking a question often overlooked in such studies: why do articles get cited at all, and why do certain papers become more highly cited than others? While there are some clearly obvious answers – writers will cite documents that are relevant to their current topic – it is important to review work that has been conducted in this area. Two studies in particular offer large-scale surveys of motives for citation: a psychology-centred study by Shadish *et al.* in 1995 [Shadish *et al.* 1995] and a 2000 paper by Case and Higgins looking at communications studies [Case & Higgins 2000]. Case and Higgins also review the previous work on citation theory and track the shifts between 'normative' theory (citation is due to relevance) and 'persuasional' theory (citation is due to self-interest), arguments that, without sufficient empirical basis, led Cronin to conclude that "it is difficult to see how citation can be defined as a norm-regulated activity" [Cronin 1984]. Adopting a practise of surveying people who cited particular papers, both studies asked the citers to choose their reason for citation from a list of 28 (Shadish) or 32 (Case/Higgins) possible reasons, basing those lists on the previous theoretical work. In both cases, responders were allowed to choose more than one reason, but had to give relative importance values for these reasons.

The results were largely consistent between the two studies (Shadish *et al.*. pp.481-2, Case/Higgins p.640). Shadish *et al.*. identified six specific citation types ("exemplar citations, negative citations, supportive citations, creative citations, personally influential citations and citations made for social reasons") while Case and Higgins identified seven ("classic citation, social reasons for citing, negative citation, creative citation, contrasting citation, similarity citation and citation to a review"). Case and Higgins further refined both their own work and that of Shadish *et al.*. by drawing out the three most significant factors for citation: "first, the perception that the work is novel, well-known and represents a genre of studies; second, the citing author's judgment that citing a prestigious work will promote the cognitive authority of his or her own work; and third, the perception that a cited item deserves criticism – which can also serve to establish the citer as an authoritative, critical thinker."

While these studies describe the major reasons for citation, it is also important to consider why certain articles are more highly-cited than others. From a citation-reason perspective, Case and Higgins state that while "we cannot reach definitive conclusions about the nature of highly-cited items" they do identify that highly-cited items are: "very likely to emphasize reviews of the literature on their topic, be cited as 'concept markers', and be authored by widely-recognized authorities in a field of research."

In addition to these methods, other empirical studies have found interesting features of highly-cited articles. Two conclusions in particular stand out. Firstly, Lawrence identifies that articles available online are cited on average 336% more than those not available online [Lawrence 2001]. Secondly, and related, McVeigh notes that Open Access journals are increasing in number such that "over 55% of the article content indexed by Thomson ISI in 2003 was produced by a publisher that allows some form of author-archiving." [McVeigh 2004]

Considering all the above, it is not unreasonable to suggest the following idea, central to this theses: not all citations are equally valuable when determining research influence, and they should not be treated as such. Since citations occur for a variety of reasons, and may vary according to factors such as online availability as much as scientific quality, there is clearly an argument, at the very least, for additional measures alongside the Journal Impact Factor covering such contextual information. Indeed, Garfield himself admits that Journal Impact Factors are "controversial" and notes that "the literature is replete with recommendations for corrective factors that should be considered, but in the final analysis subjective peer judgment is essential." [Garfield 1996].

While it is hard to disagree with that statement, many authors have questioned the validity of even using Impact Factor measures at all. [Dong *et al.* 2005] discuss the important point that some disciplines have longer citation half-life times than others (a metric fully discussed in section 2.1.4 below), meaning lower impact factors for journals in fields such as radiology. They also discuss the fact that cross-discipline citations are more common in some fields than others, and impact factors are often skewed as a result of this. Going a little further, Seglen [Seglen 1992] [Seglen 1994] [Seglen 1997] notes that there are a number of very good reasons not to use journal impact factors for evaluating research under any circumstances:

- Use of journal impact factors conceals the difference in article citation rates (articles in the most cited half of articles in a journal are cited 10 times as often as the least cited half).
- Journals' impact factors are determined by technicalities unrelated to the scientific quality of their articles, such as a correlation between article length and citation rate.
- Journal impact factors depend on the research field: high impact factors are likely in journals covering large areas of basic research with a rapidly expanding but short lived literature that use many references per article.

Opthof, unlike Seglen, argues that impact factors can be legitimately used to judge the research impact of journals but draws a number of very clear restrictions on the use of this metric, most notably that journal impact factors may not be legitimately applied to individual papers, authors or groups of scientists (such as research groups or institutions) who produce fewer than 100 papers in the JIF-standard two year period of measurement [Opthof 1997]. This observation may also legitimately be extended to smaller disciplines producing fewer overall papers: the two-year window is clearly not suitable for all fields of science.

Beyond the statistical validity or otherwise of journal impact factors, there are other considerations that need to be taken into account. For instance, it is worth noting that in today's online age, 'citation lag' is shortening and thus the two year standard may not be the correct timescale on which to judge citation impact, although given the varying frequency of journal publications, it may not be meaningful to reduce the standard to below a two year figure [Kleinberg 1999].

Additionally, there are arguments against the types of citations used in impact measures. Seglen notes that "self citations are not corrected for" [Seglen 1997] in journal impact factor measurement, leading to self-inflation; while Gabehart points out that articles later retracted by journals are frequently positively cited due to there being no method of tying a retraction item to the original article in citation analysis [Gabehart 2005]. Further, Dong *et al.* note an inconsistency in the differentiation between 'citable' and 'non-citable' articles in the Impact Factor calculations: while the denominator may not include 'non-citable' articles such as letters and editorials, these may be counted in the numerator of the equation, leading to an inflated impact factor when non-citable items are actually cited [Dong *et al.* 2005]. All the above contribute to the "controversy" [Garfield 1996] surrounding the use and abuse of journal impact factors, and the debate continues.

Beyond the direct criticisms of the Impact Factor approach to research evaluation, there are deeper philosophical ideas that need to be taken into account. Kuhn's critique of the structure of scientific revolutions challenges the notion of smooth development of scientific ideas, instead suggesting a model of jagged changes, often requiring paradigm shift as part of this development [Kuhn 1970]. As an example, Kuhn describes Einstein's 1905 Special Relativity paper as a classic example of a 'paradigm shift' paper: one that not only allowed a jump in scientific development, but in fact opened up an entire new sub-branch of science by introducing a new way of thinking. In terms of citation counting, the Special Relativity paper is less cited than Einstein's 'Brownian Motion' paper of the same year, but may be considered more influential as it opened a new sub-discipline of physics. While citations are clearly important – indeed, Special Relativity is still one of the highest-cited papers in history – a pure citation count model does not take into account the importance of turning-point papers, papers producing critical changes in the development of science.

Additional problems arise when taking the citation count results beyond the level of the paper. The logical extension of paper citation figures is to add them together for a particular author and calculate the total or mean number of citations their papers receive as a way of determining their overall influence, thus producing a set of results such as Citeseer's 'most cited authors' list. However, there are clearly some problems with this approach: in particular, Citeseer's list ranks D. Johnson as the most-cited author in Computer Science, despite the fact that there are 26 potential D. Johnsons within Citeseer's database (last name Johnson, first name begins with D) and Citeseer's creators admit that the list even includes non-D. Johnsons such as Joel T. Johnson [Han *et al.* 2005]. Despite this, as stated above, the list is widely viewed and downloaded, and indeed used in research: the author ranking by number of citations that resulted from Citeseer data is highly correlated with that obtained from ISI/JIF [Zhao 2005].

2.1.3 Relative citation scores might be an improvement

There is clearly, therefore, a demand for alternative methods for determining influence of papers, authors, institutions and even journals. While citation counting is, and will remain, the primary method for determining whether work is relevant ("normative" theory accounts for the largest sub-group of reasons for citation in the studies by both Shadish *et al.* and Case and Higgins), a number of alternative methods have been suggested and applied. Jon Kleinberg quotes a 1976 study by Pinski and Narin [Pinski & Narin 1976], noting their "more subtle citation-based measure of standing, stemming from the observation that not all citations are equally important. They argued that a journal is 'influential' if, recursively, it is heavily cited by other influential journals." [Kleinberg 1998] However, such algorithms are computationally expensive and perhaps unrealistic as a practical alternative to traditional impact factors, at least until recently.

Kleinberg draws connections between Pinski and Narin's work and his own hyperlink analysis algorithm for determining hubs and authorities on the web. While noting that document purpose is different in the two fields, and thus weightings in the algorithms will be different, there are clear parallels in the processes involved. Indeed, the large-scale citation network engine Citeseer "aims to identify hubs and authorities in the scientific literature" [Lawrence *et al.* 1999a] by applying Kleinberg's techniques to its current corpus of over 700,000 documents. However, while Kleinberg's algorithms do provide a computationally-realistic implementation of the ideas presented by Pinski and Narin, they have yet to be accepted as standard by the wider research community. Part of the aim of this thesis is to show the value of a hubs-and-authorities model for influence ranking over a straight citation count model, and will be covered empirically in chapter five.

Other models have been proposed that might provide alternatives to the pure citation count approach. Google's patented 'PageRank' model [Page *et al.* 1998] for web page relevance rankings based on link analysis draws partly from Kleinberg's work and may be considered another alternative. Shadbolt, Brody, Carr and Harnad, in considering the future of Open Archiving and the potential for future metrics, summarized fourteen 'candidate' measures that could be applied, including hubs/authorities, citation counting and other non-bibliometric measures such as downloads [Shadbolt *et al.* 2006]. While the validity of some of these measures will continue to be questioned, it is worth considering measures beyond the boundaries of pure bibliometrics.

2.1.4 Other possible measures: beyond bibliometrics

As noted in section 2.1.2, [Dong *et al.* 2005] showed that the lifetime of journal citations varies across different disciplines. This observation has been expressed in more formal terms by Sombatsompop *et al.*: citations tend to peak and decay according a half-life model [Sombatsompop *et al.* 2004]. Similar to the half-life curve generated by measuring decaying radioactive material (as defined in the 'Definitions and Abbreviations Used' section), citations to a paper peak followed by a steady decay curve. As such, peak latencies and half-life measures can and should be used to compare paper citations in a like-for-like manner: a comparison of two papers should not be based on how many citations each paper has received, but how many citations they have received at the same point on the citation half-life curve. This key observation, also noted by [Shadbolt *et al.* 2006], shows the importance of context-based metrics beyond pure citation counting.

As well as PageRank and Hits (Kleinberg's Hubs and Authorities), other graph analysis models have been proposed in various contexts such as the JUNG framework described in section 2.5.2. Of these, a key metric used in citation analysis has been Betweenness Centrality [Brandes 2001]. This is, in essence, a measure showing how important any single node is to the graph overall by measuring how many possible paths of the graph pass through each node: if a relatively large number of possible cross-graph paths pass through a single node, the node must be relatively important to the overall graph. In terms of bibliometric citation graphs, a node with a high degree of Betweenness Centrality will be one that leads to new areas of science developing, papers which may be regarded as 'turning points' in the Kuhnian sense. Chen's CiteSpace viewer [Chen 2004] [Chen 2006] applies Betweenness Centrality algorithms to small groups of citation-linked papers, and along with other algorithms such as Pathfinder [Schvaneveldt 1989], Chen uses this to produce a viewer (according to year or sub-discipline) which scores papers according to citation degree and Betweenness Centrality, and shows the progression of scientific thought in that particular sub-discipline. Figure 2.1 shows a chronological view of the social network analysis discipline, with large nodes representing key turning-points in the discipline.

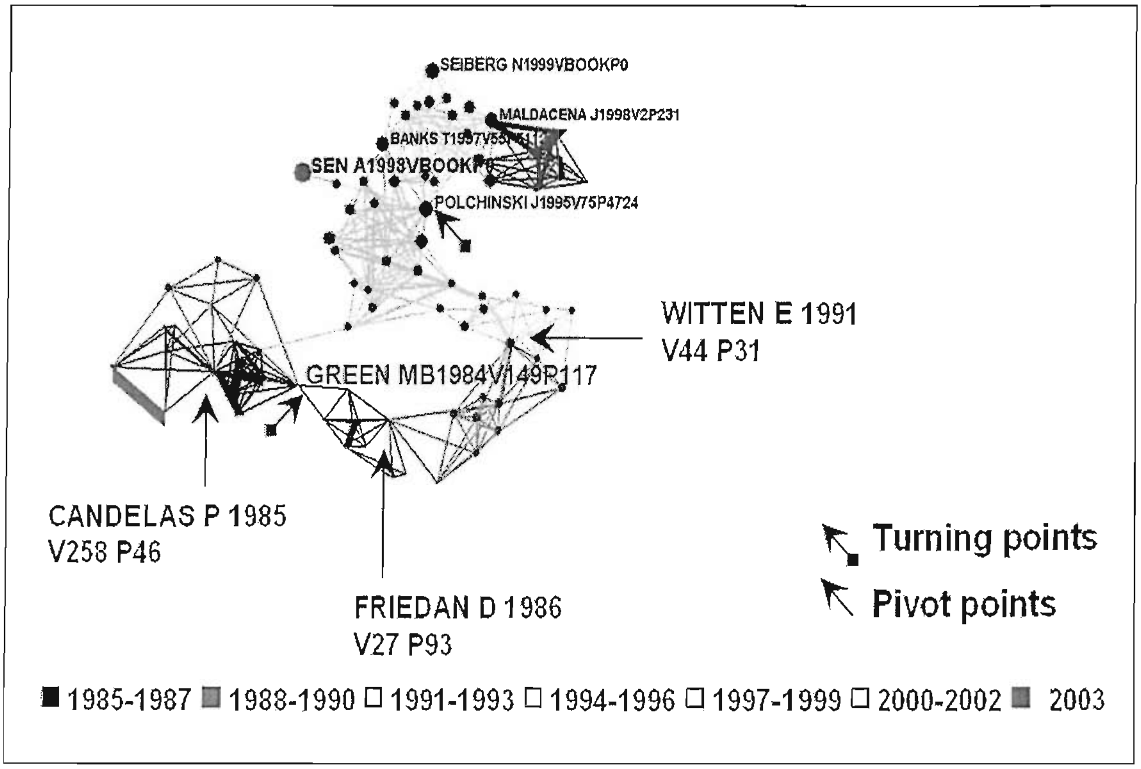


Figure 2.1 : Chen's Citespace showing discipline development over time.

Also noted by [Shadbolt *et al.* 2006] is the use of download statistics as a predictor of overall citation rates. Although obviously restricted to documents available online for downloads, this measure has been shown [Bollen *et al.* 2005] to accurately predict both citation totals and the citation curve and half-life referred to above. As the number of papers available for download increases due to the boom of both online digital libraries and personal/institutional repositories, this statistic can increasingly be used as a predictor of citation-based influence. However, once the citation curve itself does begin, the citation rates should take over as the prominent measure: downloads should only be used as a predictor rather than an alternative measure to be taken alongside citation counts.

Moving away from individual paper influence scores, a further measure that can be applied at the levels of author, group or organisation is that of acknowledgement analysis. Councill and Giles [Councill & Giles 2004] [Councill *et al.* 2005] showed, through comparisons with Citeseer citation count analysis, that the rates of acknowledgements received from within papers closely mirrors that of citation rates for authors, groups and organisations and therefore acknowledgement analysis can be shown to be "at least as useful" [Councill & Giles 2004] as citation counting for determining research influence.

Weale *et al.* took a different approach to the question of what to do with the Journal Impact Factor as currently produced [Weale *et al.* 2004]. Identifying that part of the problem was that journals are rated according to the overall number of citations received by the individual papers, they questioned the legitimacy of using the total citation count for a journal to rate that journal when it may just be one or two particularly outstanding articles that bring up the citation rate of an otherwise ordinary publication. Instead they discuss the possibility of using rates of non-citations as a means of producing inverse impact metrics: if a journal is particularly unsuccessful, then none of the papers within are receiving citations and thus it is safe to say the publication is lacking in impact. Highly-cited journals (which is to say, journals containing highly-cited papers) *may or may not* be highly impactful. However, journals which are lacking *any* highly-cited articles are clearly exhibiting low influence levels and thus can be seen to be ineffective. As a result, Weale *et al.*. argue that high citation rates show little, but low citation rates across an entire journal prove its ineffectiveness.

As was mentioned in sections 2.1.1 and 2.1.2, the desire to produce a simple measure for rating authors based upon citation numbers has led to the use of Citeseer's 'author statistics' page being highly viewed and research being conducted into the statistical validity of the list [Zhao 2005]. Logical extensions of the citation count model allow for the counting of total number of citations received by all papers written by a particular author, or calculating the mean number of citations per paper written, and lists such as Citeseer's usually use one or other of these approaches.

An alternative that can be applied at the level of authors was specified by [Hirsh 2005] as the 'H-Index'. This measure is designed to remove the problem of high citation numbers simply by virtue of high publication rates (the problem of total citation counting over a career) while also avoiding the inherent problem in calculating mean number of citations per paper, which allows less influential papers to drag down the score of an author who may have published several highly influential papers. The H-Index is calculated by counting the number of papers written (n) and d etermining how many of those n papers have received >n citations.

Straightforward to calculate (given correct citation counts), it allows authors to determine their influence over a career as they require both a high number of papers and consistently high citation rates across many papers in order to produce a high H-Index score. Using this measure, Hirsh showed a correlation between Nobel prize winners and high H-Index scores, thus arguing strongly in favour of its validity as a statistical method; however, he also noted variances in H-Index scores across different disciplines and acknowledged the need for discipline weighting in normalizing H-Index scores across all scientists. A solution to this inter-disciplinary problem has yet to be proposed at the time of writing this thesis as it raises questions such as the health or even worth of disciplines as well as simply calculating 'normalising' mathematical factors.

While simple and backed up by evidence, criticisms and improvements to the H-Index model have emerged since the initial paper was published. [Bjelobrk & Zukerman 2005] argue that the H-Index is inherently biased against newer researchers and thus a more legitimate measure is to divide the overall H-Index by the number of years since the first paper was produced. Their report shows a comparison of the two H-Index methods against a ranking produced by expert analysis, the results of which favour their new method. [Egghe 2006] proposes a G-Index based on the number of papers which together contain $>n^2$ citations, thereby giving greater weight to very highly cited papers, which the H-Index does not. [Roussaeu 2006] suggests the joint use of H- and G-indices in determining author influence. Other criticisms levelled at the H-Index include its failure to account for citation half-life and the question of whether the total number of citations received is in fact legitimate, since Hirsh, like the original Journal Impact Factor, uses ISI's statistics to produce results, and thus the problems with ISI (such as treating all citations with equal weight) are applicable here also. A further argument against the H-Index deals with its treatment of all citations as being equal: as has been argued in section 2.1.2, all citations are not equal. This argument is dealt with in more depth in chapter five of this thesis and an alternative 'Modified H-Index' proposed, which takes relative citation importance into account.

Building on the successful models of Citeseer, Google Scholar and other crawler-populated digital libraries, a new digital library named Rexa was released by the University of Massachusetts in 2006 [McCallum 2006]. However, unlike others of its type, Rexa aims to produce legitimate influence scores for more than just papers: the system treats papers, authors and topics as first-class objects, allowing analyses to be conducted on them. Metrics include the now-familiar citation counting and H-Index scores, but also addressed issues of topic diversity and cross-pollination between disciplines, using automated topic classification techniques to allow cross-topic citation analyses to be performed.

However, the bottom line is that while many diverse influence scores have been proposed at many levels of granularity, it is clear that as yet no clear alternative to the traditional Journal Impact Factor and even pure citation counting has been adopted. Perhaps most interestingly is the means by which the proposals suggested in this section have been tested: almost all have been tested for their validity against some kind of real-world expert analysis, whether that means asking local experts for their opinion [Bjelobrk & Zuckerman 2005], comparing against the assessments of Nobel prize judges [Hirsh 2005] or contrasting results with a research assessment exercise [Harnad *et al.* 2003]. Indeed, [Garfield 1996] states that the bottom line in research influence assessment is the degree to which experts in a given field consider a particular piece of research to be influential, and stresses the importance of peer reviews – and peer review scores, usually unavailable – as part of this process.

While the work presented in this thesis draws directly or indirectly from all the material covered above, and similar tests performed using expert judgement as a benchmark, it is clear there is no immediately forthcoming panacea answer to the question of determining research influence. Instead the emphasis should be placed on a usable, flexible framework which allows different weightings and metrics to be performed on a variety of data, allowing research influence measures to be performed over large-scale datasets with minimal complexity.

2.2 Large-scale Digital Libraries

One of the requirements central to the possibilities of producing even simple bibliometrics is the need to have a collection of papers, or at least their metadata, collected together for analysis to be performed. The ISI Journal Impact Factor is only feasible because of the collection of journals they were able to consider and draw together. Additionally, the traditional means of storing journals and conference proceedings has been through institutional libraries, with the possibility of borrowing from other libraries as necessary.

The online equivalent of these collections of papers, journals and proceedings has come to be known as 'Digital Libraries' (DLs). This is a catch-all term, encompassing academic paper stores, online journals and business-oriented Document Management Systems. Appendix A contains a comparative list of commercial and non-commercial DLs investigated as part of this research. For this summary, however, it is important to note three particular types of academic DL: institutional archives, publisher's paper stores and crawled search-engine-style collections.

Institutional Archives

Not exactly a direct equivalent of the institutional library, institutional repositories are online stores where members of an institution can archive their publications, making them visible to the world (including search engines) in a standard format with standard metadata. In particular, a study by [Lawrence 2001] showed that articles available online tend to be cited much more than offline-only articles (on average by 336%), and this has led to the development of standard institutional repository software. Two systems in particular stand out as field leaders at the time of writing this report: [EPrints] and [DSpace]. The systems allow for documents, including multiple revisions, to be submitted by authors and reviewed and released as appropriate by administrators. Metadata such as Dublin Core and OAI is made available and downloading of the paper permitted. While there remains an ongoing debate over copyright issues of articles in expensive journals, the move towards archiving, often mandated by the institution, seems to be gathering pace [Shadbolt *et al.* 2006]. Standardised institutional digital libraries are therefore becoming increasingly common, and will continue to do so.

Academic paper stores

In addition to institutional repositories, archiving can also be performed by publishers and by specialists across particular fields. An example of the former is the Association of Computing Machinery (ACM) Digital Library [ACM DL] which contains the full text and metadata of every article published by the ACM. The entire record – containing journal articles, conference proceedings, book chapters and whole books – totals over seven hundred thousand documents and is additionally linked via the [CrossRef] system to forty-five additional publishers, providing links to over 6.5 million documents. An example of the latter is [arXiv], a physical sciences e-print archive active since 1991, which pre-dates online journal articles and allows authors to deposit both published and pre-print (often pre-refereed) material.

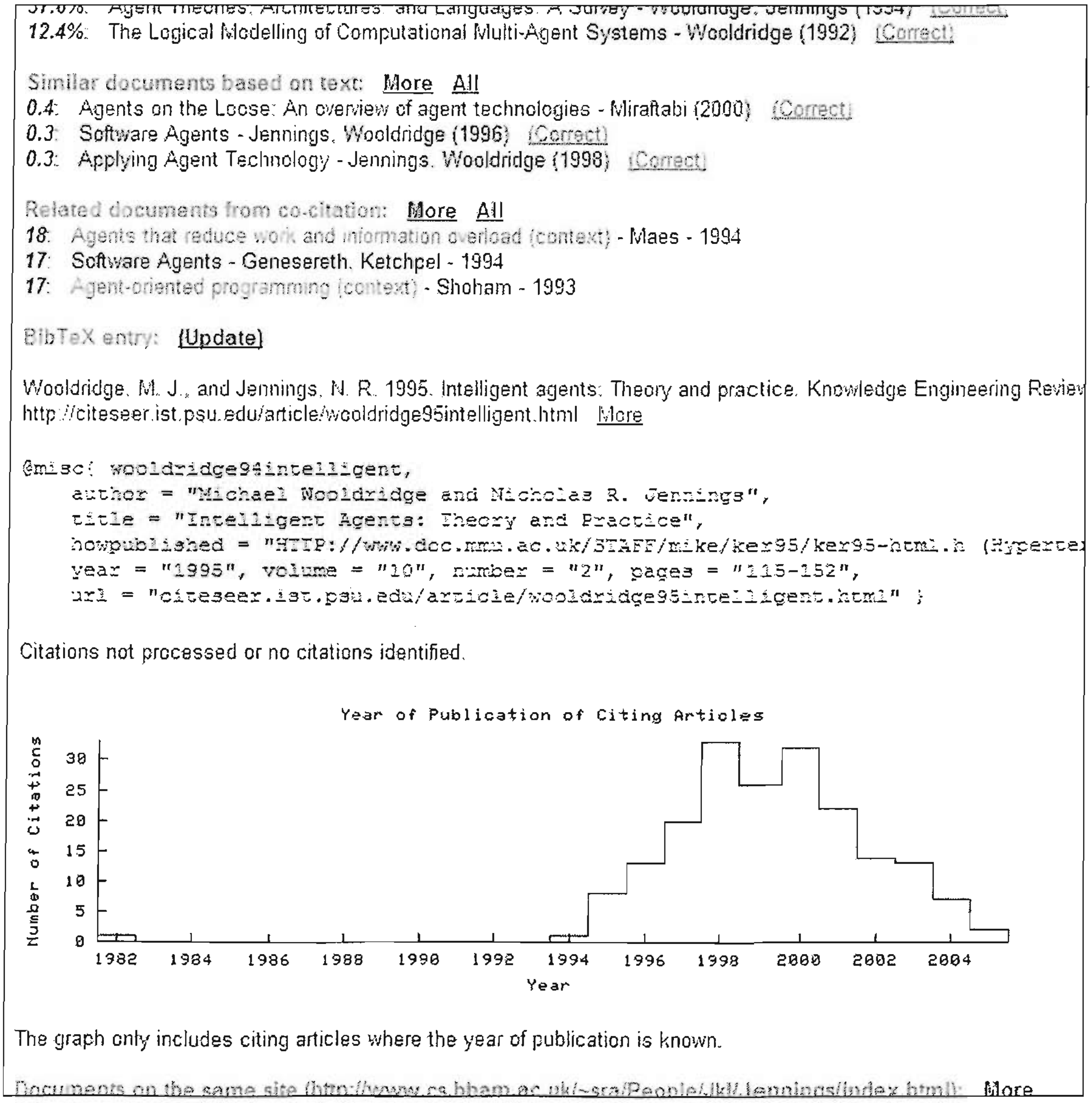
Crawlers

The final type of DL is, in some senses, not a library at all, but a search engine populated by a web crawler. Initially developed in 1997 by NEC, Citeseer [Lawrence *et al.* 1999] was the first example of this, where a crawler was developed to download freely-available academic papers, focusing mainly on the field of Computer Science. The text was extracted and processed to bring out metadata information such as title, authors, and the bibliography – which was then matched with titles of other papers in the database to provide a graph of citation links. While this provides an interesting set of metadata, the search facility built on top of the database, along with the stored cache of papers that had been crawled, meant that about 95% of the hits Citeseer received were people searching for papers and downloading them. While more general search engines such as Google would also be crawling and indexing the same papers, Citeseer gives the advantage of restricting the search to computer science papers, as well as providing access to a cached version.

Citeseer's remarkable growth led to similar search services being developed such as [Google Scholar], [MSN Academic Live] and, as mentioned in section 2.1.4, [Rexa]. While not technically developed as repositories, these facilities cache papers, expose them on the web and provide search facilities to find them, and as such may be considered digital libraries. Within the scope of this thesis, the metadata held and produced by these crawler-based stores is certainly considered as useful as that of any other type of digital library, the only difference being its automatic extraction from the paper rather than its manual entry at the point of submission.

2.2.1 Measuring Citations

In addition to the metadata produced by these scraper-based DLs, a number of them share Citeseer's feature of citation linking. This feature, termed Autonomous Citation Indexing, not only allows users to access cited papers via a hyperlink but also allows graph analysis techniques to be performed on the overall citation graph. Among those described in section 1 of this chapter, Citeseer performs citation counting and hub/authority calculations [Lawrence *et al.* 1999a], as well as producing a graphical representation of citations over time (see figure 2.2); however it must be noted that only citations from papers contained in the DL will count towards these totals, raising questions of how representative and statistically-relevant the paper base in the DL is.



37.0%: Agent Theories, Architectures and Languages: A Survey - Wooldridge, Jennings (1994) (Correct)
12.4%: The Logical Modelling of Computational Multi-Agent Systems - Wooldridge (1992) (Correct)

Similar documents based on text: More All
0.4: Agents on the Loose: An overview of agent technologies - Miraftabi (2000) (Correct)
0.3: Software Agents - Jennings, Wooldridge (1996) (Correct)
0.3: Applying Agent Technology - Jennings, Wooldridge (1998) (Correct)

Related documents from co-citation: More All
18: Agents that reduce work and information overload (context) - Maes - 1994
17: Software Agents - Genesereth, Ketchpel - 1994
17: Agent-oriented programming (context) - Shoham - 1993

BibTeX entry: (Update)

Wooldridge, M. J., and Jennings, N. R. 1995. Intelligent agents: Theory and practice. Knowledge Engineering Review
http://citeseer.ist.psu.edu/article/wooldridge95intelligent.html More

```
@misc{ wooldridge94intelligent,
    author = "Michael Wooldridge and Nicholas R. Jennings",
    title = "Intelligent Agents: Theory and Practice",
    howpublished = "HTTP://www.doc.mmu.ac.uk/STAFF/mike/ker95/ker95-html.h (Hypertex
    year = "1995", volume = "10", number = "2", pages = "115-152",
    url = "citeseer.ist.psu.edu/article/wooldridge95intelligent.html" }
```

Citations not processed or no citations identified.

The graph only includes citing articles where the year of publication is known.

Documents on the same site (http://www.cs.bham.ac.uk/~sra/People/Jkl/.Jennings/index.html): More

Figure 2.2 : Citeseer's citation count graph for a typical paper

Similarly, other citation-linking services such as Google Scholar and Rexa perform a variety of measures for each paper in their base. Of particular interest is recent work by Rexa, mentioned in section 2.1.4 above, where calculations were performed using their paper base of a little over 300,000 papers to reveal citation-based statistics not only for papers but also at the author, journal and topic level, with measures for the last of these including cross-pollination statistics showing the spread of citations across different sub-disciplines within a field [McCallum 2006]. While useful, the number of different DLs available, each with its own coverage of a particular discipline, means it is hard to say for certain just how many citations a given paper has received.

In addition to the scraper-based DLs, others also perform citation analysis leading to citation counts and related metrics becoming available. The ACM Digital Library, for instance, includes all citation links from papers within ACM publications (periodicals, proceedings, book chapters and whole books). However, as with other DLs, the questions of coverage (as well as overlap with other DLs) remains. Citation counting metrics, and all the associated metrics discussed in section 1, are clearly in demand: the growing need is for methods of confirming accuracy and reliability given so many competing DLs and their differing opinions.

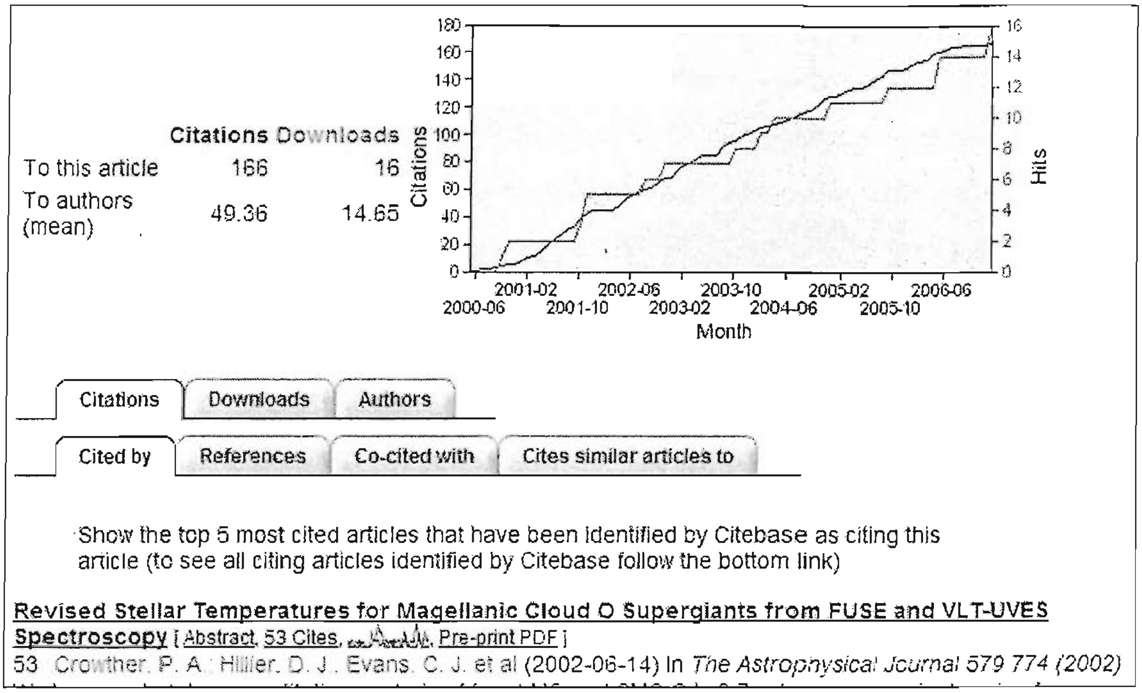
2.2.2 *Move to Open Archiving*

The move towards Open Archiving can therefore be considered as both a positive and a negative step. The Open Archive Initiative [OAI] in particular exists as part of a move to promote interoperability standards to help content dissemination, and encourage participation in two ways: firstly to encourage data providers to publish their data in OAI-PMH (Protocol for Metadata Harvesting) format, and secondly encouraging service providers to build services based upon the results of OAI-PMH requests, the OAI themselves publicising the services in return. While providing a standard format for metadata publication, which is clearly a positive step, it remains the case that querying times, availability of data and overlap of papers between repositories remains a problem, along with the question of citation linking between repositories.

However, as Open Archiving gathers pace [McVeigh 2004] it is clear that the overall percentage of papers appearing in some kind of online DL with some kind of metadata publishing will continue to grow and in many fields will soon reach a percentage such that the vast majority of papers in a particular discipline will be available in this way. Whether the noise from the variety of sources drowns out the meaning that can be gleaned from the data is instead the question that will need addressing.

To a certain extent, OAI-PMH attempts to answer this question. [Dublin Core] has become a metadata standard in both commercial and academic DLs, and is central to OAI-PMH. Defining an independently-derived set of metadata fields applicable across a wide variety of documents, Dublin Core allows users (or automated text extractors) to populate a database with a standard set of information, and systems to be built that could use such information. With such standards in place and adhered to by OAI-PMH data providers, services can realistically begin to be built to perform a variety of calculations and visualisations on the data concerned. However, the question of co-reference resolution across multiple data sources remains a problem for the service provider: while Dublin Core attempts to provide hints through fields such as Title and Source, and many DLs perform some degree of internal disambiguation it remains the job of the data aggregator to sort through data, tying together and de-duplicating as necessary.

An example of a provider building services over multiple datasets is Citebase [Brody 2003]. Although wisely noting that it remains a demonstration and should not be used for academic research, the system currently allows searching over a number of OAI-PMH sources on a variety of fields including paper title, author, publication title and keywords from the paper abstract. In addition it provides services such as lists of citations by other papers, download statistics, co-citation links and others. Figure 2.3 shows the download/citation correlation graph derived from these statistics for a typical paper. Again while facing the issue of multiple sources leading to co-reference resolution issues and de-duplication, Citebase does show the potential – and the clear demand – for federated, mediated search and metadata facilities operating over a variety of sources. As the percentage of papers made available through open archiving increases, however, it is clear that services such as those pioneered by Citebase are only the beginning of the story.



	Citations	Downloads
To this article	166	16
To authors (mean)	49.36	14.65

Show the top 5 most cited articles that have been identified by Citebase as citing this article (to see all citing articles identified by Citebase follow the bottom link)

Revised Stellar Temperatures for Magellanic Cloud O Supergiants from FUSE and VLT-UVES Spectroscopy [Abstract, 53 Cites, Pre-print PDF]
53 Crowther, P. A.; Hillier, D. J.; Evans, C. J. et al (2002-06-14) In *The Astrophysical Journal 579 774 (2002)*

Figure 2.3 : Citebase statistics for a typical paper.

2.3 The Semantic Web

While the Web as we know it allows for dissemination and cross-referencing on a scale never before known, the documents placed on the web are typically intended for human, rather than computer consumption. This means the documents are written with visual clues for the user to be able to tell, for example, what is a title and what is an abstract. Part of the aim of the Semantic Web [Berners-Lee *et al.* 1998] [Shadbolt *et al.* 2006a] is for documents to encapsulate their information in machine-readable format, such that their contents can be processed by other computer programs (typically software agents performing specific tasks), allowing reasoning and inferencing to follow.

One central feature of the Semantic Web is the Resource Description Framework (RDF) [McBride *et al.* 2004], which is an XML-based language allowing data to be expressed as subject - predicate - object triples. The subject and object are both URIs (although the object may also be a literal value), therefore allowing definitions to be referenced from anywhere on the web. Notably, the predicate is also a URI, allowing relationships to be considered as objects in other triples. This means that when assertions are made, inferences can be drawn by investigating the relationships between both objects and predicates. Specialist databases have been created to store data held in the form of RDF triples: known as triplestores, these systems have been created with a variety of capabilities according to the needs of problem domains. Of particular note are Jena [HP 2003], which provides a powerful inference engine, and 3Store [Harris & Gibbins 2003] which provides a scalable solution capable of storing and searching millions of RDF triples.

One problem that quickly emerges, however, is that of representation of similar concepts in different domains. For instance, US zip codes are, in many ways, conceptually similar to UK postal codes, in that they represent a set of postal delivery points, although the actual instances could never map to one another due to being used in different geographic domains. Although they are represented differently in the different domains (eg with different names) and from some perspectives, they aren't the same thing (zip codes cover whole towns, postal codes cover an average of fifteen delivery points), there are clear similarities between the concept of 'UK post code' and 'US zip code', which may or may not be relevant depending on the domain in question. Comparison requires knowledge of common meanings or mappings between them, according to the perspective and context the user is coming from.

Ontologies begin to answer this problem. Ontologies are, in the domain of Artificial Intelligence and Computer Science, formal definitions of relationships among terms. Gruber [Gruber 1993] described ontologies as being the "explicit specification of a conceptualisation". Essentially, an ontology defines classes of objects and their relationships in a specific domain, leading to the possibilities for inference and the potential for powerful reasoning systems.

The Web Ontology Language, OWL [McGuinness & van Harmelen 2004], is designed to provide features to describe these ontology relationships on top of the RDF structure. The features provided by OWL "include hierarchical and restrictive subclassing, transitive, inverse, symmetric and functional properties, equivalence and disjointness, cardinality, and data typing." [Millard 2004] Expressing ontologies in OWL format allows simple dissemination of ontologies over the web, as well as allowing for the extension of existing ontologies and mapping between classes and instances. OWL is designed to promote the re-use and re-purposing of ontologies, allowing the discovery and use of knowledge on a global scale.

While OWL and RDF, along with triplestores, provide the data format and storage, a further question relates to querying such data sources. The SPARQL query language [Prud'hommeaux & Seaborne 2006] provides a standard interface to such data sources, allowing extraction of information from RDF graphs in the same way that SQL provides a standard interface to traditional relational database management systems (RDBMSs). The development of SPARQL as a standard for both local and remote querying of triplestores completes the toolset required for the creation of semantic web applications.

The central feature of the Semantic Web, therefore, is the turning of information into knowledge and allowing global access to that knowledge. Built on a foundation of RDF and ontologies, this model allows rich data sources to be mapped, merged and inferred such that global knowledge can be examined, mined and investigated in a way never before possible: where the computers, not the humans, do the 'eyeballing' to check data, answer complex questions and make discoveries.

2.3.1 Application to Citation/Digital Library culture

While the Semantic Web is a vision covering a wide range of applications, the interest of this thesis is seeing how Semantic Web Technologies could be applied to online DLs. Although it is immediately clear that RDF presents a potential alternative as a method of exposing metadata, the more important question is whether there would be any value in doing so. With the OAI becoming a standard metadata format for web harvesting, and service providers able to build services using metadata thus harvested, it raises the question of whether RDF-based metadata stores are a nice idea, ultimately redundant.

A more important question, however, is to consider the development of the Semantic Web as a whole and begin to ask what the role of online DLs will be within that emerging framework. Already Citeseer's next generation development "CiteseerX" brings attention to the importance for DLs of providing a "service oriented architecture" which targets semantic agents [Petinot *et al.* 2004]. These agents, whether web crawlers or more sophisticated targeted agents, will be seeking RDF triples as part of their work. Therefore in creating semantic DLs and services it is clear that either data stores will have to output their metadata in RDF (raising the question of which ontology the data will be asserted against and whether a standard is feasible, as with Dublin Core and OAI) or the service providers, aggregating the OAI-harvested data, will need to expose their data (perhaps including the raw harvested data) in RDF, asserted against a suitable ontology.

2.3.2 Use of ontologies in academic paper research

For the purposes of the experiments described in this thesis, a standard research ontology dealing with papers, people, groups and institutions has been required. The work in this thesis has been sponsored and performed within the Advanced Knowledge Technologies consortium, which is a six-year EPSRC-sponsored Interdisciplinary Research Collaboration running from 2000-2006. Based across five UK universities, the project researched a variety of knowledge and artificial intelligence areas, including the Semantic Web and ontologies. Drawing from experience within the AKT consortium, the [AKT Reference Ontology] was chosen as a suitable data structure, extensible as required. This ontology was designed as an all-encompassing tool for investigating research, specifically covering the domain of academic research people, papers and projects, and the relationships between these. The ontology has been successfully used in semantic web applications such as OntoCOPI [Alani *et al.* 2002], a social network application for calculating 'communities of practice' and CS AKTive Space [schraefel *et al.* 2003], a semantic browsing tool for viewing Computer Science research within the UK. The ontology required slight extensions in order to become fully useful in the work described in this thesis, and these extensions are covered in detail in Appendix B. Essentially, the extended ontology contained classes and relations such that paper details (including research area) could be held in a standard format, searchable via SPARQL queries

2.4 Knowledge Reuse and Mapping Knowledge Domains

Having established the background of the need for effective measuring of paper influence, the globally-scoping possibility of large-scale digital libraries and the ontological reasoning the Semantic Web looks to provide, the question arises of how these three research areas could be brought together to achieve the aim of quantifying scientific research and its value. This section considers existing research in these areas, considering in particular the need to focus on mapping domains of knowledge and patterns of scientific research and collaboration. In each of the following sections, a specific research area focusing on the task of mapping scientific domains is summarised, followed by an explanation of the relevance of the work to this thesis.

2.4.1 Coauthorship networks and patterns of scientific collaboration.

[Newman 2004] studies co-authorship networks in three specific domains: biology, mathematics and physics. The paper concludes that by looking at the network distance between individual authors, it is possible to determine social and professional interlinking between scientists and how that changes over time and across different disciplines.

While noting the work of others in this area, the significance of Newman's work to this report is the suggestion that such network analysis services could run over the top of a variety of corpora (such as the Citeseer corpus and a network of EPrints archives) showing communities at varying grainsizes, over time and across international boundaries. While such co-authorship information is of interest in and of itself, it also provides part of this report's method for author disambiguation, which will be presented in chapter 3. It should be noted that a co-authorship network analysis tool developed by AKT, OntoCOPI [Alani *et al.* 2002], already provides a lot of the functionality described by Newman's article and its principles have guided the development of the author disambiguation tool.

2.4.2 Characterizing PNAS: Impact Maps

Using the ISI/JIF method of impact measurement according to citation count, [Boyack 2004] shows 'impact maps' that reveal concentrated areas of research and how they change over time. The paper also introduces the idea of acknowledgement analysis: "very few papers exclude a funding acknowledgement inadvertently" and thus the links between funding and individual papers is clear. The results show that higher-cited documents come from more highly-funded projects and tend to have a larger number of authors, and also reveal that concentrated areas of research can later branch out into other areas (such as core gene work expanding into cancer, RNA, cloning).

The key areas of interest to this report are the study of influence changing over time, the introduction of acknowledgement analysis and the visualisation of impact maps, even though those impact maps use citation count for influence measurement rather than the richer methods described in this report.

2.4.3 Mapping topics and topic bursts in PNAS

Using textual analysis to determine highly frequent words and topics, and burst-detection algorithms (based on Kleinberg's work in this area [Kleinberg 2002]), Mane and Börner [Mane & Börner 2004] describe their method of creating co-word-space maps. The purpose is to show how new topics emerge and how frontiers of scientific development change over time.

Although again relying on pure citation count as a measure of influence and using that, along with "expert feedback", to determine their initial data and techniques, the research described is relevant to this report in the area of using burst-analysis techniques (where words 'burst' into the literature and become widely used over a short period of time) to determine the development of scientific disciplines. This work in itself provides a new method of influence analysis, richer than one that uses citation count alone.

2.4.4 Visualizing a knowledge domain with cartographic means

Based on traditional cartography methods ("a science dealing with the transformation of spatial information") and utilizing large-format visualization, [Skupin 2004] presents a framework for studying different clustering techniques. It is these clustering techniques, in particular a term-dominance landscape allowing semantic zooming, that is of interest to this report.

The ability to graphically display a 'map of science' is a theme running throughout these articles, but Skupin's paper has been particularly influential on this thesis in two key areas: his assertion that there is no single best way to partition a domain (and thus multiple views and overlay techniques are required), and his combination of term-dominance landscape with k-mean solution to allow what he terms 'semantic zooming': the ability to zoom in on closely-clustered domains thus providing a truly scalable domain/sub-domain map. Figure 2.4 shows an example of such a map. Additionally, the term 'semantic zooming' has been augmented in this thesis to include the transit of ontology-based data to 'zoom' to various levels of granularity such as paper, author and institution levels.

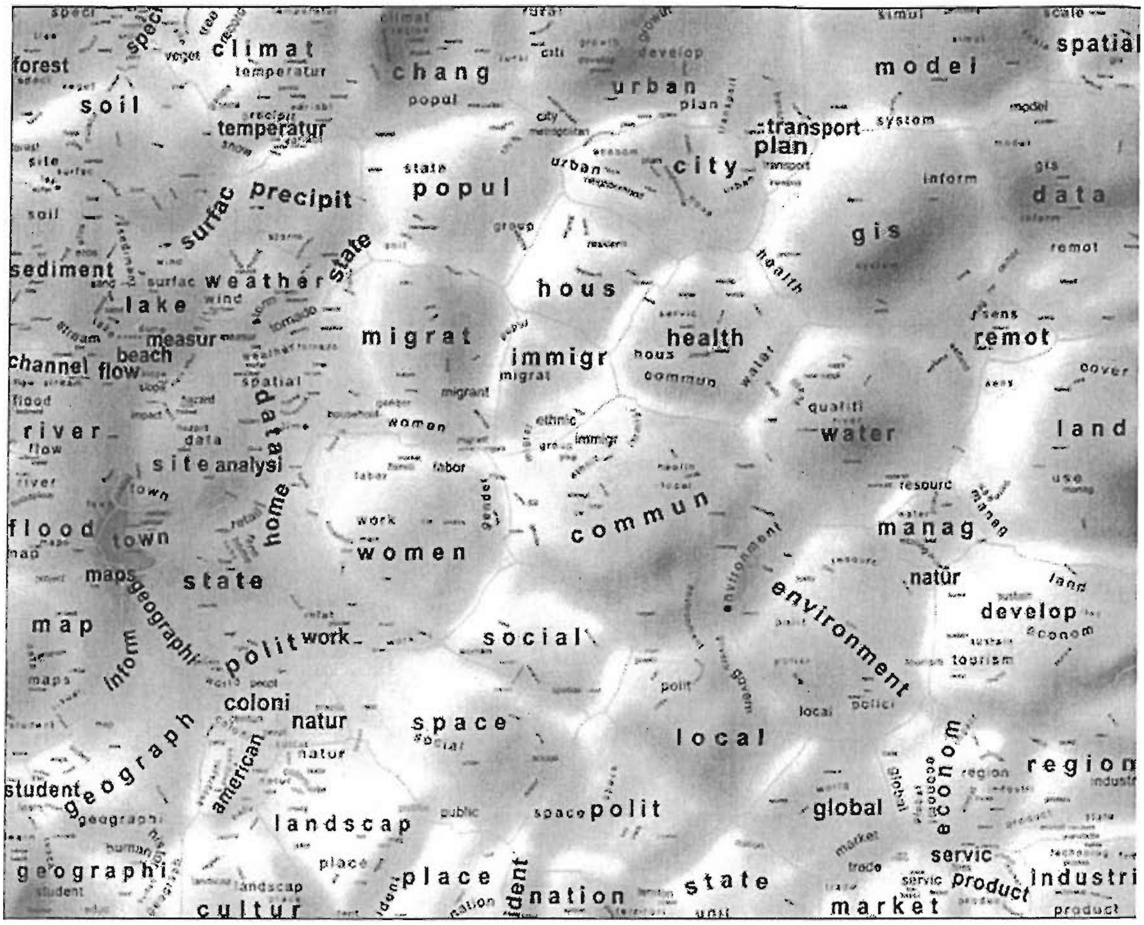


Figure 2.4 : Skupin's geographical representation of a term-dominance landscape

These papers, while the result of independent research, show both the 'state of the art' in knowledge mapping research, and suggest the direction in which things are headed. As such they have influenced the approaches taken in the research described in this report.

2.5 Metadata Transformation and Preparation

While the above sections have detailed the problem domain and influential research, there are several other areas concerning metadata preparation that require background information to set the scene for the work described in this thesis.

2.5.1 Identity Uncertainty

One area of particular interest is that of identity uncertainty and co-reference resolution. Often ambiguous data is available such as author names or institution affiliations and the question is how best to disambiguate these items of information to produce coherent, joined-up data. There are a number of approaches that have been suggested in the literature, approaches varying according to context and suitability.

String-based analysis such as Levenshtein's String-Edit distance [Levenshtein 1965] can be used to determine the differences or 'distance' between two strings. Specifically, the Levenshtein distance is the minimum number of individual character changes (additions, subtractions or replacements) required to turn one string into another.

Commonly used in natural language processing, a variety of similar algorithms exist, many of which have been included in SimMetrics [Chapman 2004], a resource for string comparison tools. SimMetrics was chosen for use in the work described in this thesis where string-based analysis was required as it both contained a comprehensive variety of string similarity measures and it normalised all results to a scale from 0.0 to 1.0, thus allowing comparison of measures if required.

A different methodology towards disambiguation is the graph-based based approach where nodes are joined together by edges of varying lengths, and disambiguation can be performed by analysing the edges to identify similar nodes that may be matches. [Malin 2005] describes such a system, using link analysis to disambiguate names found within natural language text. In the context of the Semantic Web, [Alani *et al.* 2002a] produced a general RDF referential integrity tool based on graph analysis techniques. The novel author disambiguation system detailed in chapter three of this thesis similarly uses a graph-based approach to solve the problem of identity uncertainty.

While both the above methods are largely manually-overseen processes, machine learning techniques have also been widely used in the area of identity resolution. In particular, naïve Bayes approaches and Support Vector Machines have been widely used in this area, for example by [Han *et al.* 2004]. While these approaches require large training sets, their overall effectiveness and success is clear and comparable with manually-overseen techniques.

In addition to the methods described above, canonical lists help the disambiguation process by giving a definite list of potential solutions. An example of this is in the area of academic institution disambiguation, where within the UK a canonical list of all institutions is held by the Higher Education Statistics Agency [HESA]. While a global list currently does not exist, efforts such as [Braintrack] and [HEIR] are attempting to draw together national lists to produce a canonical global list of institutions. Given the widespread demand and expectation for such services, particularly in the Semantic Web community, it is not unreasonable to expect a tool such as Braintrack to produce, in time, such canonical lists.

2.5.2 Data Manipulation Techniques

Large-scale datasets such as metadata sets for digital libraries often require specialist techniques for performing data manipulation and calculations. In addition to the standard RDBMS and Semantic Web RDF/ontology approaches to data storage, two additional tools are relevant to the work described in this thesis and were used in the experiments described in the following chapters.

[JUNG], the Java Universal Network/Graph Framework, is one of a number of potentially useful graph-analysis tools. It was chosen for use in the experiments described in the following chapters as it contained a number of graph analysis techniques useful to the semiometrics process, including a variety of node-importance algorithms such as Kleinberg's hubs-and-authorities algorithm, Google's PageRank and Betweenness Centrality as used by Chen in the CiteSpace application. Another reason for choosing JUNG over other network analysis tools such as [UCINET] was the flexible API that allowed the importance calculations to be embedded into the applications described in this thesis.

Linked in to the JUNG framework is the [Pajek] net format. Pajek is a fully-operational graph-analysis tool in itself, and incorporates a simple text format for data storage. In implementing the applications covered in the following chapters, the design choice was made to store data in simple Pajek net format before importing into the JUNG framework and performing the required importance calculations. This allowed the data to be stored in a standard format, potentially for use in other applications including Pajek, and proved an efficient method of representing large-scale citation networks. Appendix B describes the technical details of the influence calculations within the overall system architecture.

This chapter has summarised the background areas covered in this thesis, along with the essential tools and algorithms that will be used in the remainder of this thesis. The following chapter described an initial data manipulation experiment, detailing a novel technique for author disambiguation using graph analysis algorithms.

Chapter 3

Author Disambiguation

This chapter describes AKTiveAuthor, an implementation of a novel approach to solving the author disambiguation problem described in section 2.5.1. The desire for definitive data and the semantic web drive for inference over heterogeneous data sources requires co-reference resolution to be performed on those data. When considered in the context of the work described in this thesis, which is concerned with the amalgamation of data at different levels, the need for such data disambiguation is clear. In particular, author name disambiguation is required to allow accurate publication lists, citation counts and influence measures to be determined. This chapter describes a novel graph-based approach to author disambiguation on large-scale citation networks such as Citeseer's metadata set. Using self-citation, co-authorship and document source analyses, the AKTiveAuthor application is introduced, which clusters papers into groups without the need for either an existing canonical list against which to compare or supervised machine learning techniques, such as Naïve Bayes or Support Vector Machines as described by [Han *et al.* 2004]. The results are analysed and compared with other systems from the literature, and the chapter concludes that not only is such a system required to prepare data for the experiments described in the following three chapters, but that the AKTiveAuthor system, along with its self-citation approach to paper matching, is the best available means of performing this data preparation, thus proving the first sub-hypothesis of this thesis.

The technical implementation details of AKTiveAuthor are covered in Appendix B. AKTiveAuthor is described and evaluated in [McRae-Spencer & Shadbolt 2006].

3.1 The AKTiveAuthor Problem Domain

As automated information extraction systems become increasingly common, there is an increased demand to know whether two similar names refer to the same real-world object or not. This is observed in place names (San Jose is the capital of Costa Rica and also a city in California) and academic institutions (an affiliation to an academic institution named “Southampton” could refer to the University of Southampton or the Southampton College that is part of Long Island University, USA). This phenomenon is particularly problematic when considering author names of research papers or bibliography citations. Two specific problems exist. Firstly, one author may have multiple aliases, such as Professor Nick Jennings appearing in various citations and papers as ‘Nicholas Jennings’, ‘N. Jennings’ and ‘Nick R. Jennings’. Secondly, multiple authors may have a similar or even identical name, such as David L. Harris (Professor of Engineering at Harvey Mudd College, formerly with Stanford and MIT) and David L. Harris (Infrastructure Systems Engineering Department, Sandia Labs, Albuquerque).

The scale of this problem can be seen simply by considering Citeseer’s own ‘Author Statistics’ page [Citeseer Author Statistics], where, as described in section 2.1.2, ‘D Johnson’ is given the status of being the most cited author in Computer Science. In reality, there are 26 distinct D Johnsons within the Citeseer dataset, thereby calling into question the validity of the statistics. Consider also the potential for multiple aliases – for example, B Croft and W Croft appearing separately on the list despite W. Bruce Croft being one single person – and it quickly becomes clear there is a need for some kind of disambiguation process to differentiate between real-world authors if these statistics are to be of any use as justifiable, valid metrics.

3.2 Overview of the AKTiveAuthor system

AKTiveAuthor presents a novel approach to the problem of automated name disambiguation in the specific context of a large-scale citation network. Autonomous Citation Indexing, described in section 2.2 of this thesis, provides a citation graph that links papers (and their metadata) together according to their bibliographies. The approach described in this chapter is centred around the observation that within these citation graphs, there is a tendency for authors to cite their own previous work. Sample testing showed that when papers cite work by an author with the same last name, roughly 95% of the time it is the same author. This approach can be used to iteratively tie together papers within a citation graph to eventually yield a collection of papers that should be by the same author. Figure 3.1 shows a partial citation graph for papers authored by Nick Jennings, using a subset of data collected from Citeseer for papers between 1992 and 1999. The graph details papers linked by their bibliographies. It is important to note that no other author with the last name Jennings cites, or is cited by, any of these papers. The central feature of the approach presented in this paper is the use of this citation graph between authors sharing the same last name to yield groups of papers that are all by the same author.

While the self-citation observation described above may yield results with an accuracy of around 95%, figure 3.1 also shows that the different graphs created may not entirely link up. This is consistent with researchers' practice: often researchers will have once major area of interest and will develop that work over time with successive papers, but will often have one or two minor areas of interest which yield perhaps only one or two papers that would include bibliographic references to each other.

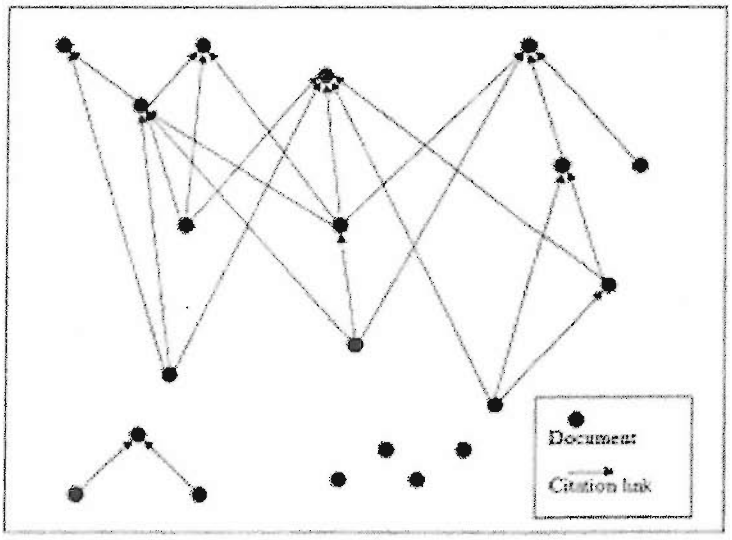


Figure 3.1: Partial citation graph for Nick Jennings (sample data 1992-99). Shows one major group of citation-linked documents, one smaller group and four papers that are not linked by the author's citation graph.

For instance, figure 3.1 shows a large group of thirteen papers authored by Nick Jennings on the subject of agent-based computing, and a smaller group of three papers which deal with the subject of economics. While the author is the same person, the papers from the two different areas do not cite each other at any point, which is understandable given that they are two different strands of work in two largely unrelated areas. The Jennings citation graph thus reveals that while there is only one author, Jennings does seem to have two 'research identities': one in the area of agent-based computing and one in the area of economics. While this is an interesting observation in itself, our aim is to produce a complete record of all Jennings' publications, whether they are to do with agents, economics, or anything else. However, this observation of 'research persona' identified through citation analysis is sufficiently interesting and potentially useful in contexts such as expert-finder applications, that further research into this area should be performed.

In order to overcome this barrier to a more complete picture of Jennings' work, two metadata attributes are used to further tie together these graph fragments. Firstly, the use of co-authorship analysis is widely acknowledged as a method for both author disambiguation [Malin 2005] and determining social networks [Alani *et al.* 2002], and is here applied in much the same way as [Han *et al.* 2004]. Secondly, the Citeseer metadata set also includes a field called 'url' (equivalent of Dublin Core's 'source' field) which gives the exact URL of the file when it was harvested by Citeseer. By cutting off the final filename part of the URL, we have a URL which will match any other files held on the same site. For a large part, Citeseer harvests from personalised (rather than institution-wide) repositories and as such, other papers with the same 'url' base authored by someone with a similar name will usually be by the same person. This approach tends to group an author's work according to time – as people move from institution to institution, old repositories (normally lists of hyperlinks on web pages) are generally left and not updated, while the author's new institution will create a new web page for that author. Combining this chronological-grouping method with the subject-grouping effect of the self-citation analysis, and tying in the general-usage co-authorship algorithm, the Jennings example is reduced down to one large group of papers with just one outlying paper unconnected in any of the three ways described to any of the other papers. Such sample observations have driven the larger-scale experiments using the same approach described in the remainder of this chapter.

3.3 Methodology and experiments

While the above section summarises the three main methods (self-citation, co-authorship, source URL) used in our analysis, there are actually five steps to the over-all AKTiveAuthor process. The first step deals with initial clustering of 'possible matches' from the entire database. The second, third and fourth steps apply the three approaches described in section one. It is worth noting that the order in which these steps are applied makes no difference to the overall outcome of the process: the order given is simply the order in which they were implemented. The fifth step performs a final 'sanity name-check' before committing to linking two papers and asserting that they are by the same author.

3.3.1 *Step one: Initial clustering*

To test the effectiveness of the method, it is necessary to check the results against real-world data, which means checking by hand. While the experiments described in the following three chapters require merged data formed from the data on an entire digital library (such as Citeseer), for the purposes of experimentation it is necessary to break down the work into sets of papers small enough that they can be checked by hand. Apart from a small number of mis-spellings on papers or mis-reads by Citeseer's parser, the last name of the author is very accurately held by the Citeseer database. As such, the analysis for each author need only be performed using the cluster of papers whose author have that last name. For example, to find all the papers authored by Nick Jennings, it is only necessary to look at the cluster of papers whose author list contains the name 'Jennings'. This observations therefore also allows by-hand checking for each name-cluster.

For the purposes of this experiment, eight name-clusters were chosen, ranging from relatively rare names (Glaser, 79 papers in Citeseer) through to very common (Johnson, 2201 papers in Citeseer). Also included were names with a wide spread of different authors (Harris, Hall) and names that, due to the nature of Citeseer, are heavily weighted in favour of one particular author (Giles, Lawrence). This wide-ranging choice of names would therefore allow us to determine if our method favoured any particular type of data. Further, future experiments should take into account more name-clusters, however due to the requirement of by-hand checking of results it was not practical for the purposes of this experiment to perform anything other than boundary testing of common/uncommon and spread/skewed name-clusters.

The experiment therefore begins with the clustering from the database of all the papers authored by someone with the last name being tested. These are stored in an array for use by the program but also written out to a file, which is then used in the manual disambiguation process for checking results later. The eight by-hand disambiguated files are then stored for use as a benchmarking resource for future experiments.

3.3.2 Step two: Self-citation analysis

The first pass at tying the name-cluster papers together is to apply the self-citation graph. Initially, each paper is put in a collection (in the case of this implementation, a vector) of size one, containing only the paper itself. Each paper in the name-cluster is successively tested against every other paper to see if the second paper is in the bibliography of the first, or vice versa. If it is found that the two papers are linked by a citation relationship, the collection associated with the second paper is added to the collection associated with the first. It is important to note that this is not a probabilistic approach: in all three steps, the collections are augmented on a straight yes/no decision of whether the similarity measure has been found. Over time, these collections grow and the collections will eventually resemble the groups shown in figure 3.1. In each case, the 'sanity name-check' of step five is applied before committing to the change.

3.3.3 Step three: Co-author analysis

The second pass at tying papers together is to apply the co-authorship analysis to the papers. This is currently the least rigorously-applied of the three methods: at present, documents are linked together on the basis that they are co-authored by authors with the same last name. For example, papers 334113 ("A Classification Scheme for Negotiation in Electronic Commerce") and 5494 ("Pitfalls of agent-oriented development"), both papers in the Jennings name-cluster, also have a co-author with the last name Wooldridge. In this (and most) cases, the Wooldridge in question is the same person. However, in some cases this approach will lead to a small number of incorrect matches, particularly in papers where the authors have particularly common names or where papers have a large number of authors. For the most part these are cleared up by step five, the 'sanity name-check', but a few may still slip through.

3.3.4 Step four: Source URL analysis

The final pass at tying papers together involves linking papers that were scraped from the same website. While this is a Citeseer-specific piece of metadata, the Dublin

Core 'source' field may mean that other citation network data sources can have this step applied to them too. Frequently only a small number of papers are stored at the same URL (often only five or six), but this information can still often link together disparate collections created in steps two and three.

3.3.5 Step five: Sanity name-check

While it is not possible to say that two people who share a name are the same person (for instance, manual checking reveals nine distinct David Johnsons within Citeseer's dataset), it is certainly feasible to suggest that two people with different names may be considered different people: Norman L. Johnson is different from David E. Johnson in almost every conceivable case. Step five of the process invokes this observation. Before committing to tying together two authors (or two author 'collections'), the full names are checked against each other to see if they are obviously not the same person. The criteria for determining a non-match aren't totally obvious: for our program we considered a conflicting initial letters as a sign of a non-match, along with conflicting stemmed names. For instance, authors such as Earl and Erik Johnson would not be merged as their names conflict, while Nicholas and Nick Jennings would be merged due to their stem (in this instance, the first three letters of the name) being the same. Future iterations of this algorithm should include a gazetteer of name-stems and equivalences to allow for more subtle matches to be made.

3.4 Metrics

Despite the potential shortfalls in the methods described above, the results described in the following section are extremely encouraging. Before considering these, however, it is important to explain the nature of these results. Unlike [Han *et al.* 2004], we are considering the results from the point of view of the real-world authors rather than from the collection of test papers we are looking to classify. As such, a straight accuracy measure of 'how many papers did we match with the correct canonical author' does not work: we are looking to create 'canonical authors' as part of the process. Our results therefore more closely reflect information retrieval work and yield three scores: d-precision, d-recall and d-f-measure, novel metrics described in depth in the following sections. For evaluation purposes, the results section of this chapter does, however, contain a comparison with two existing machine learning approaches described in the literature.

3.4.1 D-Precision

D-Precision is defined as the proportion of relevant documents of all documents retrieved:

DP = (number of relevant documents retrieved) / (number of documents retrieved)

In the context of our study, it is relative to each individual paper. Each paper ends up in a group (of size at least one) and that paper will have a d-precision that reflects how many papers in its group should be there. For instance, a group may contain four papers, three by one author and the fourth by a different author. In this case, the d-precision for three of the papers is 0.75 (3/4 are relevant) and for the fourth 0.25 (1/4 are relevant). The arithmetic mean d-precision for this group of papers can therefore be calculated based on the fact that three papers have a d-precision of 0.75 and one has a d-precision of 0.25: ((0.75 x 3) + (0.25 x 1)) / 4 = 0.625. Mathematically, this can be reflected as follows:

For one paper, d-precision = $|A_r| / |A|$, where A is the number of documents returned and A_r is the number of relevant documents returned.

For one group as identified by AKTiveAuthor:

$$\text{d-precision} = (\sum_{n} (|A_r|^2) / |A|))/ A$$

- where n is the number of papers in an individual group linked together by AKTiveAuthor.

3.4.2 D-Recall

D-Recall is defined as the proportion of retrieved documents of all relevant documents available:

DR = (number of relevant documents retrieved) / (number of relevant documents)

In the context of our study, as with d-precision, d-recall is relative to each individual paper. Each paper will have a d-recall figure that reflects how many relevant documents are in the same group as that paper. For instance, a group may contain four papers, all of which are by the same author, but there is also a fifth paper by that author that has not been linked to any other papers at all. In that instance, the four papers in the group have a d-recall of 0.8 (4 out of 5 relevant papers are present) and the paper on its own has a d-recall of 0.2 (only 1 out of 5 papers are present). The arithmetic mean for this author's overall set of documents can be calculated from the fact that four out of the five documents have a d-recall of 0.8, and one has a d-recall of 0.2: ((0.8 x 4) + (0.2 x 1)) / 5 = 0.68. Mathematically, this can be reflected as follows:

For one paper, recall = $|A| / |A_{rtot}|$, where A is the number of documents returned and A_{rtot} is the number of relevant documents in total for a given author.

For one real-world author group:

$$\text{recall} = \Big(\sum_{rtot} (|A_{rtot}|^2) / |A_{rtot}|\Big) / A$$

3.4.3 D-F-measure

The F-measure is a widely-accepted combination of precision and recall discussed in detail in [Van Rijsbergen 1979]. It is calculated as the harmonic mean of precision and recall. Similarly, these experiments introduce a D-F-measure, reflecting the harmonic mean of d-precision and d-recall, ie:

(2 x (D-Precision x D-Recall)) / (D-Precision + D-Recall)

The harmonic mean is chosen as it prevents skewed data rating highly, instead favouring data where both d-precision and d-recall tend to be higher. For example, if d-precision is 0.9 and d-recall 0.7, the arithmetic mean is 0.8 but the D-F-measure is 0.7875. However, if the d-precision and d-recall are both 0.8 (the arithmetic mean remaining at 0.8), the D-F-measure is also 0.8. In the context of this chapter, the D-F-measure is calculated after the overall d-precision and d-recall figures have been calculated for each name-cluster.

3.5 Results

Figure 3.2 shows the overall results for the AKTiveAuthor system against the eight chosen name-clusters. It is important to note that authors who have written exactly one document are not included in these results: they produce an automatic result of 1.000 for both d-precision (unless they are pulled into another author's collection, which is very rare) and d-recall. By not including these, the overall results are lower but show more clearly the effectiveness of the system when testing different parameters as set out below.

Surname (size of cluster)	**D-Precision**	**D-Recall**	**D-F-measure**
Carr (242)	1.000	0.754	0.860
Giles (414)	0.998	0.935	0.965
Glaser (79)	1.000	0.824	0.904
Hall (644)	0.996	0.783	0.877
Harris (477)	0.992	0.705	0.824
Jennings (389)	1.000	0.852	0.920
Johnson (2201)	0.991	0.806	0.889
Lawrence (353)	1.000	0.883	0.938
Average (Arith. Mean)	**0.997**	**0.818**	**0.899**

Figure 3.2 : Results for the eight name-clusters, including the sample size and the three metrics for each name along with the overall result.

3.5.1 D-precision higher than d-recall

The first thing to note is that the d-precision is consistently much higher than the d-recall. This is in line with expectations as set out above: self-citation will lead to very high d-precision (the sanity check in step 5 will increase this further) but will not draw in documents that are outside an authors main citation network. Additionally, authors with more than one major area of research interest tend to end up with more than one main citation network, a feature which leads to particularly low d-recall results. This shows that the ongoing challenge with author disambiguation is to increase d-recall without losing the very high d-precision scores provided by self-citation analysis. An additional interesting note is that papers that remain outliers, keeping d-recall down, tend to be those with few overall citations and lower authority scores. In the overall context of the work described in this thesis, particularly the work concerned with producing overall influence figures for authors and institutions, it is worthwhile noting that in almost all cases, outlying papers that will therefore not be contributing to overall influence scores are those which add little to the data anyway. Finally, it is clear that these results compare favourably with others in the literature, even though direct comparison is not possible: for example Machine Learning techniques such as those described by [Han *et al.* 2004] output an 'accuracy' figure against a known set of results. This is different to both d-precision and d-recall, although related to both, and clearly not directly comparable with the D-F-measure. However, the 0.899 average D-F-measure would appear to out-perform the Naïve Bayes (0.733 accuracy) and Support Vector Machine (0.654 accuracy) approaches as described in the literature by [Han *et al.* 2004] as both d-precision and d-recall are higher than the accuracy figures given. The AKTiveAuthor approach described above also has the advantage of flexibility in that it does not require a training set nor a known result set against which it compares candidate matches. This approach can therefore be seen to out-perform existing approaches both in terms of flexibility and accuracy of results.

3.5.2 All three methods are required for good results

Breaking down the results according to the three matching methods shows that for the highest d-recall results, all three methods are required. As stated above, self-citation analysis alone gives a good starting point for the process: for the entire Citeseer database, self-citation links together 107 of Nick Jennings' total 277 documents into one large group, while a number of other small groups of around five documents each are also created. However, the metrics reveal that this gives a d-recall figure of only 0.154 as many small groups of documents give a low overall d-recall figure.

Adding the second method, co-authorship analysis, boosts the figure largely through joining the small groups created in the initial section. Self-citation and co-authorship analyses together link 248 of Nick Jennings' 277 documents (co-authorship analysis alone links together 214). However, a relatively large number of individual documents remain unconnected to any others and the d-recall figure only rises to 0.802.

The final method, analysing the origin of the document, gives the final d-recall figure for Nick Jennings of 0.943. 269 of the 277 document are linked together in one group (performing origin-analysis alone matches together 101 of Nick Jennings' 277 documents), another group of two documents exist along with one on its own, while a further five remain outside the main group only because of a parsing error by Citeseer: Citeseer's parser incorrectly picked out the name 'R. Jennings' on five papers that were actually by 'Nicholas R. Jennings'. This has the effect that while the three matching methods do join the 'R. Jennings' group to the main group, the 'sanity check' of step five does not allow 'R. Jennings' to be matched with any of Nick Jennings' aliases, including N. R. Jennings.

This pattern is repeated across almost all authors. Using all three methods of tying documents together allows a d-recall figure consistently >0.7 and averaging at 0.818. Using only one or two of these methods yields consistently lower d-recall figures (usually between 0.4 and 0.7 depending on the methods used).

3.5.3 Certain types of author are better suited to these analyses

While the Jennings name-cluster has been used throughout this paper to explain the system and demonstrate its effectiveness, it is clear from the results that certain types of author respond better than others to the three methods applied. The Jennings name-cluster is an example of one where the group is dominated by one name in particular: 277 of the 389 documents (71.2%) are authored by Nick Jennings, while the next highest contributor to the group is Jim Jennings from IBM who has 33 papers in Citeseer. Other examples include the Lawrence and Giles name-clusters, where Citeseer's co-creators Steve Lawrence (199 of 353 = 56.4%) and C. Lee Giles (338 of 414 = 81.6%) dominate the groups. In these cases, d-recall figures of 0.883 and 0.935 respectively demonstrate that name-clusters dominated by one particular author tend to yield better results. By contrast, the Harris group shows a d-recall of only 0.705, with the most dominant member of that group (John G Harris of the University of Florida) authoring only 34 out of 477 documents (7.1%) in the Harris name-cluster.

Another type of author tending to get lower d-recall figures are those who author a large number of papers that cannot be tied by any of our three methods. An example of this phenomena is Professor Peter Hall of the Australian National University at Canberra whose diverse work with a variety of co-authors all over the world has led to a corpus of papers largely unlinked by self-citation, co-authorship and largely harvested by Citeseer from the (distinct) repositories of his co-authors. This leads to a d-recall figure for Peter Hall of only 0.287. A similar, although less striking, effect is seen among academics who co-author paper with a range supervised students over a wide variety of topics. It is important that people with such profiles are not excluded from the system, and therefore future iterations of the project should look to take into account such specific factors.

Finally, the results for the Johnson (2201 total papers) and Glaser (79 papers) are consistent enough to show that size of name-cluster does not appear to be a factor either way in these analyses. It is reasonable to suggest, however, that more complete clusters will yield better results: presently it is clear that Citeseer contains only a sub-section of the total number of published computer science papers in the world (estimated at around 40% of current computer science output (Petricek, Councill, Giles 2005)), thereby reducing the effect of the matching methods described in this chapter due to incomplete citation and co-authorship graphs. As more data becomes available, including sources such as the ACM dataset used in later experiments described in this thesis, it is expected that more complete datasets (both large and small) will yield better results.

3.6 Conclusions

While the results described above are encouraging – very high d-precision and mean d-recall >0.8 – it is necessary to look at the results in the context of the data usage in order to conclude whether this process is actually useful.

The purpose behind this work to disambiguate authors is to provide a number of services based on the citation graph and document metadata held in Citeseer and other digital libraries, as described in chapter two. Some of these services would include "view my papers", "count my citations" and "calculate my influence" based on amalgamated c itation counts, authority scores and other metrics. The AKTiveAuthor technique described in this chapter has therefore been used to create a dataset which is used by the experimental applications in the following chapters to prove the remaining three sub-hypotheses of this thesis. Therefore in terms of usefulness of data for these services, the results have been highly useful, although it has also been necessary to perform additional manual disambiguation in some cases to produce the most useful data. Overall, however, the results are good enough to prove useful and therefore can be considered a success.

A spin-off from the experiments has been the creation of definitive manually-disambiguated data sets for the eight name-clusters used. These have been written up in a standard format and may be used, along with Citeseer's dataset, as bench-marks against which future disambiguation work may take place. In effect, they are the 'canonical names' discussed by Han *et al.*, and can be used in either a machine learning disambiguation context (such as that of Han *et al.*) or in a 'cold-start' approach such as the one presented in this chapter.

Beyond the services and applications described in the remainder of this thesis, based on disambiguated data, future work in the area of author disambiguation includes two main areas:

- Investigating adding to the system other methods of tying papers together, including use of institutional affiliation data and the move towards probabilistic-measures (perhaps including the use of string-similarity measures such as SimMetrics [Chapman 2004]) for research area analysis, as well as improvements to the existing steps.
- Creation of further manually disambiguated name-clusters allowing further benchmarking of future disambiguation systems.

This chapter has therefore shown the novel way self-citation graph analysis can be used to produce a disambiguated set of authors and papers, and shown the results to be superior and the system more flexible than existing approaches from the literature, thus proving the first hypothesis of this thesis.

The following chapter builds on this dataset by showing how such data, held in RDF format and asserted against a standard ontology, can be used to produce a framework for web services and applications with querying and updating functionality superior to that of traditional database methods.

Chapter 4

Ontology-Assisted Data Mediation

This chapter details the background to the semiometrics system, describing the data manipulation and mediation required to allow such a system to be created, as well as introducing the client applications. Building on the data preparation described in chapter three, this chapter considers the best approach to take in terms of creating a framework for multi-level influence-measuring services. As large-scale digital libraries become more available and complete, not to mention more numerous, it is clear there is both the need and demand for services that can draw together and perform inference calculations on the metadata produced. However, the traditional Relational Database Management System (RDBMS) model, while efficiently constructed and optimised for many business structures, does not necessarily cope well with issues of concurrent data updates and retrieval at the scale of hundreds of thousands of papers. Conversely, the growth of RDF and the increasing interest in Semantic Web technologies perhaps begins to present a viable alternative approach at a scalable, practical level. This chapter specifically focuses on contrasting semantic web technologies with the more traditional database approach in the context of producing the framework for compositional views of influence, as described above. It concludes that RDF technologies are both a scalable and performance-realistic alternative to traditional RDBMS approaches. Specifically, it shows that for relationship-based queries on open-ended large-scale metadata stores, RDF technologies can significantly out-perform traditional RDBMS approaches at both a theoretical and empirical level, thus proving the second sub-hypothesis of this thesis.

The technical implementation details of the experiments described in this chapter are detailed in Appendix B. The experiment is described and evaluated in [McRae-Spencer & Shadbolt 2006a].

4.1 Problem Domain

The emergence of large-scale online digital libraries is a feature largely welcomed by the academic scientific community. While systems such as Citeseer and Google Scholar crawl the web searching for papers, increasingly online institutional repositories (such as EPrints and DSpace) are being created, exposing their papers and metadata in a standard format. These systems are sufficiently successful to have raised the expectations of the user community: it is now the case that people expect academic papers to be findable and downloadable, fully indexed and searchable in 'Google' style; citations to other documents should be rendered as hyperlinks; metadata should be searchable and services summarising the work of an author, institution or journal/conference should be available. While the various digital libraries attempt to meet some or all of these expectations, it remains the case that the number of papers indexed and stored by these libraries is in the order of hundreds of thousands and will only increase as the move towards more open archiving (described in section 2.2.2) continues and more metadata becomes available. Producing services that run over these libraries, and perhaps even across multiple libraries, is therefore a challenge when considering the issues of search speed and query complexity.

4.2 Overview of solution

While the problem domain described above has been tackled in a variety of ways, the growth of Semantic Web technologies may provide an answer to at least some of the questions raised. The push towards more intelligent, computer-readable websites has brought to the fore the use of ontologies as a means of data manipulation and integration, and RDF as a format for data storage and transfer. While much semantic web research focuses on the development of storage techniques (such as 3 Store and Jena) as well as inference-based language standards such as OWL, it is clear that RDF-based triplestores, along with the query language SPARQL, allow a different approach to be taken to data storage and searching than that which is provided in more traditional

RDBMS models. This chapter details the theory and practice of applying the RDF technique to large-scale digital library metadata and shows how, for many more complex queries demanded by the raised expectations of services described above, data storage in RDF and querying by the standard RDF query language SPARQL provides a level of performance at least as useful as standard SQL approaches, and fast and flexible enough to provide a real option for use in online digital library services.

4.3 Motivation

The relational database model, queried by SQL, has been a standard model for data storage for many years. While optimisation and indexing techniques have boosted the efficiency of this model, it remains the case that some queries on multi-table databases remain complex even though they are easily expressible in plain language. For example, given a simple database schema for a large metadata repository, some valid queries might be: 'how many distinct authors are there in this system', 'which papers cite papers by this author' and 'what are the titles of the articles this author has written since 2002'. In SQL, these could be respectively expressed as:

```
1.SELECT COUNT(*) FROM authors;

2.SELECT DISTINCT bibliographies.MasterArticle,
 bibliographies.ArticleCited
FROM bibliographies INNER JOIN author
ON bibliographies.ArticleCited = author.documentID
WHERE author.AuthorID = 'P123';

3.SELECT DISTINCT articles.Title, articles.articleID
FROM articles INNER JOIN authors
ON articles.articleID = authors.ArticleAuthored
WHERE authors.acmID = 'P123'
AND articles.Year > 2002;
```

While the first of these queries is relatively simple, the second and third both involve inner joins, the third on a potentially very large table 'articles', raising query complexity and potentially increasing the time taken to produce a result, depending on the indexing techniques used. By contrast, these two queries can be expressed relatively simply in SPARQL, given a suitable ontology: in this case, the AKT Reference Ontology as described in section 2.3.2 was used.

```
2. SELECT distinct ?p ?c
WHERE
  {
   ?p akt:has-author <http://citeseer.ecs.soton.ac.uk/#P123> .
   ?c akt:cites-publication-reference ?p .
  }

3. SELECT distinct ?p ?t
WHERE
  {
   ?p akt:has-author <http://citeseer.ecs.soton.ac.uk/#P123> .
   ?p akt:has-title ?t .
   ?p akt:has-date ?d .
   ?d support:year-of ?y .
   FILTER (?y > 2002)
  }
```

While these queries may appear similar in terms of number of lines, the actual logic involved is far simpler in the SPARQL queries, and as will be shown in this chapter, response times can be greatly reduced. However, it is wrong to suggest that SPARQL is simply better than SQL in all cases: the first query is actually far better in SQL than in SPARQL:

```
1. SELECT distinct ?a
WHERE
  {
   ?p akt:has-author ?a .
  }
```

Despite the relative simplicity of the statement, there are two major problems with this query. Firstly, the query itself doesn't actually answer the question of 'how many' – SPARQL does not contain an equivalent of SQL's count(*) operation, and so the user (or the program making the call) would have to do the summation calculation separately. Secondly, and more importantly, this SPARQL statement has to query the entire Knowledge Base, finding all instances of the 'has-author' predicate, then creating a distinct list of the subjects of those triples. This is an extremely inefficient way to simply count all the instances of authors – however the nature of RDF means that we need to count the instances of the relationship in order to discover the identity of the URIs concerned: they are defined as being authors because they are subjects of triples whose predicate is 'has-author'. This contrasts with the RDBMS approach, where authors are defined as being authors because they appear in the authors table, and all that has to be done is to count the number of rows in that table.

4.4 Data Storage Models and Purpose

The essential difference between the RDBMS and ontology-based data models are their respective purposes. This section discusses the design rationales behind the two approaches and where the essential differences lie.

Relational databases typically deal with questions of identity, including if that identity involves calculations across tables. RDBMSs are optimised to allow efficient querying of data, data which is itemised in tables and columns according to identity. This means that in practice, queries such as retrieving the total number of authors is straightforward – it is simply a summation of the number of distinct rows in the 'authors' table. However, queries based around relationships between data are more complex – although the relational model makes these queries possible, for large-scale databases with complex tables containing several hundred thousand rows it can be very time-consuming to perform the required JOIN operations.

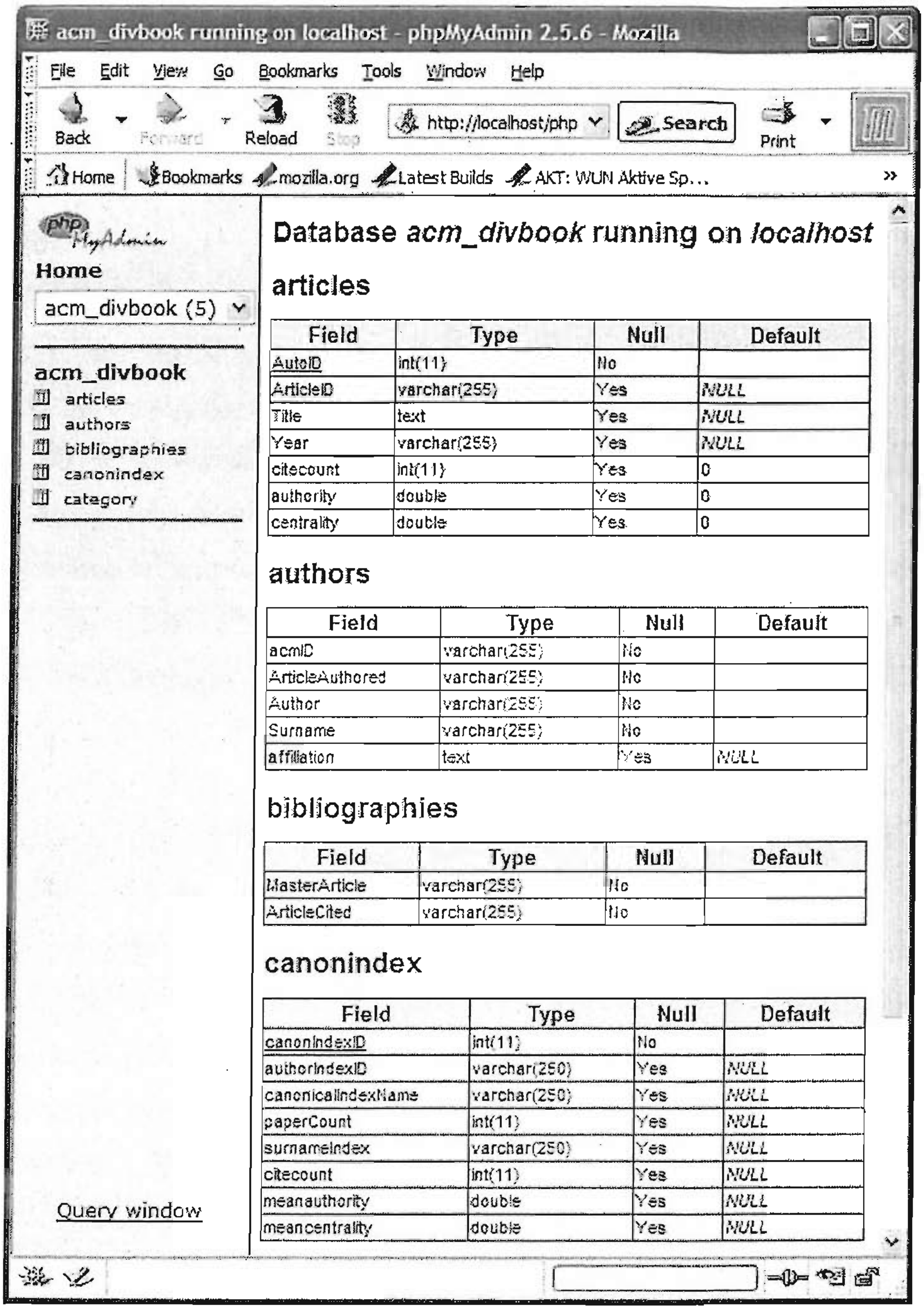
To overcome this problem, RDBMSs typically offer users the opportunity to perform indexing operations on their data. User-chosen indices allow storing of sorted columns (or column combinations) meaning a vast reduction in search time, particularly when performing the more complex relational operations. The down-side of this is an increase in the time taken to perform inserts and updates to the system, as the indices associated will have to be updated. Additionally, for large multiple-indexed tables, the index files often grow to the extent that they become bigger than the actual database files they are indexing. For most systems, a trade-off can be made between the amount of indexing and the need to keep the system 'open' so additions and changes can be made as well as efficient querying: however, as described below, as systems become larger, the trade-offs become harder to make.

In contrast to the 'identity' model of traditional RDBMS databases, ontology-based data is designed to deal primarily with questions of relationships, where the predicates are the focus of the query. The emergence of RDF as a standard format for data description, coupled with the development of scalable triplestore solutions (such as 3Store in the case of this work), has allowed the creation of searchable knowledge bases where relationship-based queries can be easily framed, provided the ontology concerned is sufficiently engineered to allow for such queries. In practice, therefore, queries such as retrieving the titles of all documents a particular author has written since 2002 is straightforward the system just needs to look for all the predicate-subject combinations where the has-author predicate is followed by the particular URI representing the given author, then filter out all results from 2002 and before. As we are essentially searching for a relationship rather than a set of answers from a table, the ontology model is suited to allow us to search for such information.

As a side-note, it is important to remember that underneath triplestores is usually a database of some description – indeed 3Store is built on top of a relational database (specifically MySQL), optimised with its own indexing. As the various experiments described in the following section compare the relative efficiencies of the SQL and SPARQL approaches, it is important to note that the SQL database used by 3Store and the one used in the experiments was the same MySQL installation on the same computer: the tests therefore were focusing not on the relative performances of databases, but on the differences between the SQL and SPARQL approaches.

4.5 Experiment Details

As stated above, the motivation for storing large-scale document repository metadata in RDF format came from the desire to produce usable, efficiently searchable services based on metadata from two computer science centric repositories: Citeseer and the ACM Digital Library. While straightforward searching and browsing facilities are fully implemented on the respective websites of these libraries, the desire was to provide more in-depth services based on data relationships, such as 'influence' scores for papers, authors and institutions based on more than purely citation counting alone. To this end, the raw metadata (essentially Dublin Core plus citations) was taken from the two sources and put into two different databases with identical schemas, as shown in figure 4.1. This schema, while containing a number of tables, was optimised to give the simplest possible view of the data in the smallest number of tables possible, while adhering to the basic relational database model. Thus there are three main tables: articles, authors and bibliographies, with a fourth (canonindex) introduced to help speed up certain author-based queries, even though this means a duplication of author data. Note: throughout these experiments there was no attempt to merge the two datasets as it was considered most useful to see how similar results would be across two completely distinct, although similarly sized, datasets.



Database *acm_divbook* running on *localhost*

articles

Field	Type	Null	Default
AutoID	int(11)	No	
ArticleID	varchar(255)	Yes	*NULL*
Title	text	Yes	*NULL*
Year	varchar(255)	Yes	*NULL*
citecount	int(11)	Yes	0
authority	double	Yes	0
centrality	double	Yes	0

authors

Field	Type	Null	Default
acmID	varchar(255)	No	
ArticleAuthored	varchar(255)	No	
Author	varchar(255)	No	
Surname	varchar(255)	No	
affiliation	text	Yes	*NULL*

bibliographies

Field	Type	Null	Default
MasterArticle	varchar(255)	No	
ArticleCited	varchar(255)	No	

canonindex

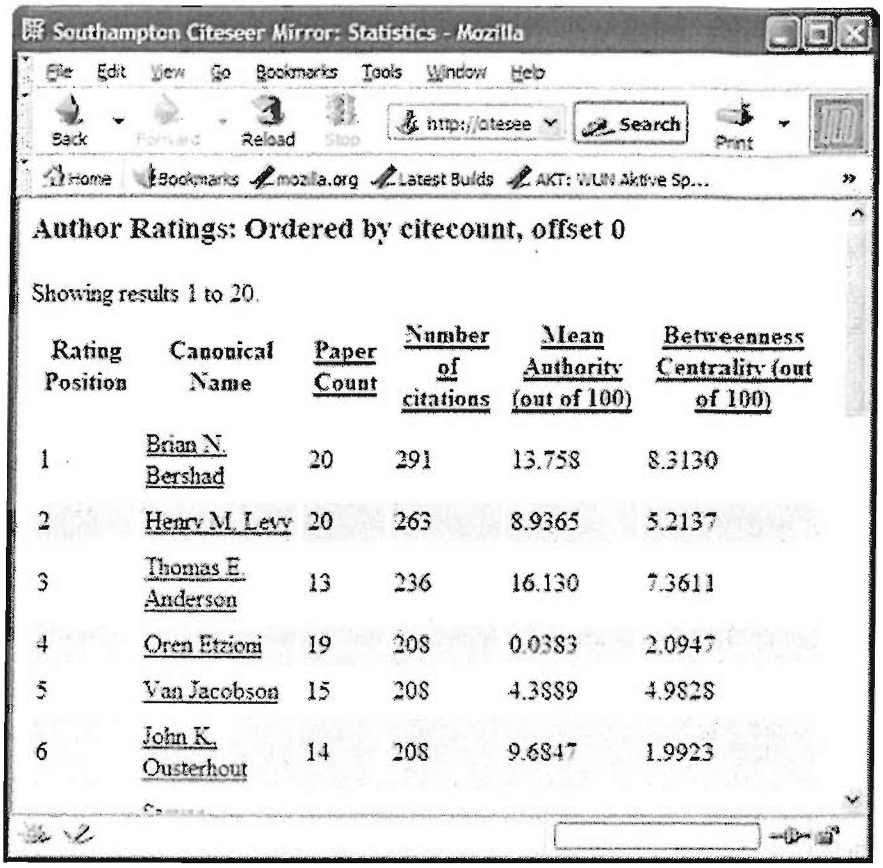
Field	Type	Null	Default
canonIndexID	int(11)	No	
authorIndexID	varchar(250)	Yes	*NULL*
canonicalIndexName	varchar(250)	Yes	*NULL*
paperCount	int(11)	Yes	*NULL*
surnameIndex	varchar(250)	Yes	*NULL*
citecount	int(11)	Yes	*NULL*
meanauthority	double	Yes	*NULL*
meancentrality	double	Yes	*NULL*

Figure 4.1 : The database schema for the MySQL sources for the ACM dataset. The Citeseer database is identical except for some different column titles such as the author ID field. Note that 'canonindex' is a short-cut index table for storing pre-calculated information on authors to speed up query times.

4.5.1 RDBMS approach

Initially questions of indexing were answered by attempting to find a sensible trade-off between the need for indexing and the need for flexibility in terms of amending and, particularly, adding data. However, it quickly became apparent that while indexing allowed for quick searches, the indexed column and table became difficult to update with new and amended data in a live environment, even if such updates were stored up and scheduled for a low-usage period. The more indices, the slower the updates, even if the tables were otherwise optimised and non-essential features (such as foreign key constraints and cascade functions) were removed from the database and handled at the application level. On a small subset of Citeseer data, containing roughly 12,000 papers, a compromise model was possible containing a degree of indexing while still allowing for changes to be made to the database. For the full datasets, however, containing metadata, author and bibliography information for over half a million papers, no such compromise was possible: either the unique columns in the tables were indexed, effectively preventing live updating, or they were not indexed, dramatically slowing search time. Eventually two models were chosen for the experiment: a 'closed' system with heavily-indexed tables that would not be updateable in a 'live' setting and an 'open' system with minimal indexing where updates could be made at the expense of search time.

Using these models the metadata, along with a few of the more in-depth results, were exposed through a number of web services. Initially implemented using the Citeseer data subset of 12,000 papers held in the 'compromise' index model described above and shown in figure 4.2, the services were expanded to the full dataset using the 'closed' model described above after the 'open' system led to more time-outs than actual results being displayed. While the 'closed' system was sufficiently quick to respond to queries, and thus useful for demonstration purposes, it was clear that in practice a system that was effectively 'frozen' would not be useful in anything other than the very short term. For the remainder of this chapter, the terms open and closed SQL databases will be used to refer to the databases produced with minimal and heavy indexing respectively.



Author Ratings: Ordered by citecount, offset 0

Showing results 1 to 20.

Rating Position	Canonical Name	Paper Count	Number of citations	Mean Authority (out of 100)	Betweenness Centrality (out of 100)
1	Brian N. Bershad	20	291	13.758	8.3130
2	Henry M. Levy	20	263	8.9365	5.2137
3	Thomas E. Anderson	13	236	16.130	7.3611
4	Oren Etzioni	19	208	0.0383	2.0947
5	Van Jacobson	15	208	4.3889	4.9828
6	John K. Ousterhout	14	208	9.6847	1.9923

Figure 4.2 : Screenshot showing the semantic web services client pages running on a sub-set of Citeseer data.

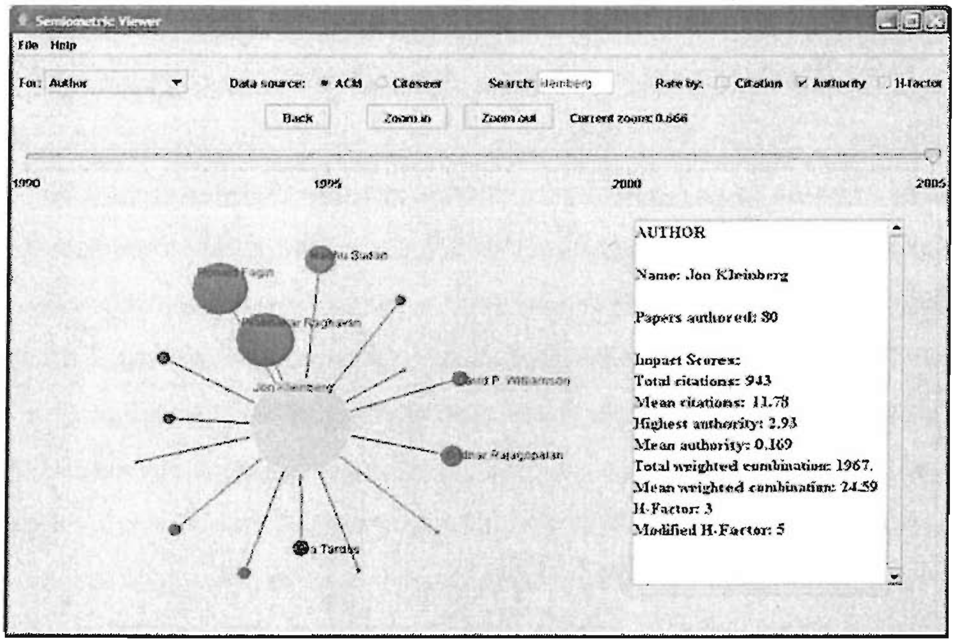
4.5.2 RDF/Ontology approach

At this point the direction of the experiment was changed to see if the RDF/ontology model could provide a solution to this problem. Although it was theoretically possible that, as described above, a SPARQL-based set of queries to a triplestore might provide a different set of response times for the same results, it was unknown whether the increase in efficiency over the 'open' SQL model would prove sufficient to be able to offer the services we wanted in a reasonable timescale. Similarly it was unclear whether SPARQL alone would be able to provide all the answers, given the examples in section 4.3 of this chapter which showed the clear advantage of SQL in identity-based queries: would a combination of SPARQL and SQL be better? The web services were thus re-written to allow a choice of SQL or SPARQL queries while conducting the experiments. In order to utilise these services, two separate client systems were created, on which the experiments could be performed.

4.5.3 Client 1: Ordered Lists and Summaries

The first of these clients was a set of web pages made available via a local server alongside a mirror of the existing Citeseer system, provided by Penn State University. These pages utilised many of the web services to provide a coherent set of "influence score" results users would be able to search and browse. With paper metadata populating the AKT Reference Ontology and with the Citeseer data augmented by the AKTiveAuthor process described in chapter 3, the services were able to query the data asserted in a 3Store running a number of other KBs, and on a server running numerous other web applications including the Citeseer mirror.

Initial results were encouraging: for the majority of searches translated into SPARQL, the searches completed in a suitable time for use in web services. The few that were too slow matched the few that SQL queries had proved capable of responding to in a reasonable timeframe from the open database. Therefore overall success of the SPARQL semantic web service querying model, combined with a small number of SQL queries to an open database, allowed the development of the second client: the creation of a Semiometric viewer application.



For: Author

Data source: ● ACM ○ Citeseer

Search: Kleinberg

Rate by: □ Citation ☑ Authority □ H-factor

Current zoom: 0.666

AUTHOR

Name: Jon Kleinberg

Papers authored: 80

Impact Scores:
Total citations: 943
Mean citations: 11.78
Highest authority: 2.93
Mean authority: 0.169
Total weighted combination: 1967.
Mean weighted combination: 24.59
H-Factor: 3
Modified H-Factor: 5

Figure 4.3 : Screenshot of SemioViewer application showing search on ACM dataset, revealing influence scores for Jon Kleinberg and the relative closeness and influence of his co-authorship community.

4.5.4 Client 2: Semiometrics Viewer

The second client, the Semiometrics Viewer, uses a combination of SPARQL and SQL queries, calculating influence scores for papers and authors on the fly, producing summary data for selected paper/author, search interface and browsing of neighbouring papers/authors (citations for papers and co-authors for authors), whose influence scores are also shown. The prototype application (shown in figure 4.3) is written in Java and calls a variety of SPARQL queries via HTTP, as well as the equivalent SQL queries directly to MySQL (querying either the open or closed database). The overall purpose of the application, in conjunction with the web services described above, is to allow the browsing and calculating of influence scores at various levels of granularity – papers, authors, institutions, disciplines and others. While the theory and results of that work is discussed in the following chapter, the SemioViewer also provides a platform for comparing the SQL and SPARQL approaches. As it contains equivalent queries in both languages, and as the application is designed to be used as a practical interface to large-scale metadata stores, it is an ideal test ground to compare equivalent SQL and SPARQL queries into data stores dealing with several hundred thousand papers.

4.5.5 Experiments

Using the two clients, the main queries were identified that are required for the systems to run. These were reduced down to a final list of thirteen key queries, eliminating queries that were essentially duplicates. The final list of queries, detailed in the results section, proved to be a mix of identity-centric and relationship-centric queries. In the actual experiments, two types of querying were used: firstly the clients themselves were used to prove the system working in practice, and secondly the queries themselves (either SQL or SPARQL) were extracted from the system and run directly on the database or knowledge base concerned to get more exact response times. In some cases, multiple queries were required (for example, getting multiple influence scores from a specific author via SQL): in this case, the time taken for the client system to respond was usually given.

4.6 Results

The purpose of this chapter, and of the experiments described here, is not to see how good the influence metrics produced by the clients are: that question is covered in chapter five. The purpose of these experiments is to ask two key questions: (1) is it useful to replace open SQL queries with SPARQL for some/all queries, particularly where SQL is very slow and (2) is it a realistic hope to produce a system that works in real time while remaining open to new data being added?

4.6.1 Summary of results

The results in figure 4.4 show the response times to a set of queries performed on the full ACM metadata set of 700,000 papers, with the tests conducted on the three methods of data storage/retrieval described above: an open database queried by SQL, a closed heavily-indexed database queried by SQL and a 3Store Knowledge Base queried by SPARQL. The thirteen tests conducted were all based on queries that either the web services or the SemioViewer application need to ask at some point in their execution.

Test #	**Test**	**Closed SQL**	**Open SQL**	**SPARQL**
1	Search for paper given incomplete title string ('AKTive').	0.83s	1.04s	23s
2	Search for author given incomplete name string ('Keller').	2.02s	2.03s	32s
3	Search for author given incomplete name string ('Johnson').	2.04s	2.04s	52s
4	Get paper details (title, year) given paper ID.	0.02s	1.28s	<0.5s
5	Get paper details (title, year, authors) given paper ID.	2.06s	3.40s	<0.5s
6	Get paper details (title, year, authors) given incomplete search string for title.	1m 34.97s	4m 0.566s	Times out.
7	Get paper's top 15 citations and relative influence scores (cite	1.27s	Times out (>30 mins)	Consistently <10s,

	count, authority, combination) given paper ID.			typically <4s.
8	Get details of author given author ID.	0.03s	0.87s	~1.5s
9	Get all papers (paper ID only) by a particular author given author ID.	0.01s	2.04s	~2s
10	Get all papers (paper ID, title, year) by a particular author given author ID.	0.78s	1m 4.03s	~3s
11	Get influences of all papers from previous test and thus calculate author influence (author has 71 papers)	1.32s	~0.5s per individual query, 37s total; times out (>30 mins) if done as one query.	<15s
12	Get influences of all papers from previous test and thus calculate author influence (author has 10 papers).	0.51s	~0.5s per individual query, 6s total; times out (>30 mins) if done as one query.	<5s
13	Get closest 15 co-authors and calculate their relative influence given author ID. (Tested on various author IDs).	Typically <10s. Maximum observed for complex author 19s.	Times out (>30 mins)	<15s for typical author. Maximum observed for complex author 27s.

Figure 4.4 : Results of experiments performed on ACM dataset using SQL and SPARQL

4.6.2 Discussion of results

The most noticeable results is the general speed advantage of the closed database: this significantly out-performs either the open database or the SPARQL triplestore queries in most cases. However realistically, while it is important for the purposes of fairness to MySQL to show the advantages of heavy indexing, the comparison in results that needs to be made is between the open SQL system and the SPARQL-based KB. There are four types of results reported in the above section: those where SQL on the open database was substantially faster than SPARQL (tests 1, 2, 3), those where open SQL and SPARQL were roughly the same and both usable (tests 4, 8, 9), those where SPARQL was substantially faster than open SQL (tests 5, 7, 10, 11, 12, 13) and those where neither open SQL nor SPARQL were quick enough to be useful (test 6). Each type of result is now considered in turn.

Open SQL faster than SPARQL

These are queries dealing with questions of identity (single table information gathering): string matching within a particular field is something SQL is heavily optimised for, even without multiple indexing. SPARQL, conversely, does not contain a 'LIKE' function and instead relies on searching all records for subjects in triples with the predicate 'has-title' or 'full-name' and then filtering on a regular expression – a much more time-consuming process. Therefore for substring queries, typically searches, it is clear that SQL is superior and should be used in a practical, real-world system. Also note that test 8 produced a marginally quicker result for open SQL than SPARQL: this is due to an optimised RDBMS searching just a single table for more identity information about a given author ID, whereas the SPARQL query has to search through a few triples to get the required information.

Open SQL and SPARQL similar

These tended to be simple queries where open SQL had to only look in a single table and SPARQL had to only find a small number of predicate-subject combinations. In practice, either query type may be used for these queries and equally good results may be expected.

SPARQL faster than open SQL

The nature of the SemioViewer application means this is the largest group of results: calculating influences and co-authorship communities requires a more intense study of relationships between data. For SPARQL, this is ideal: it has been optimised for searching object-predicate or predicate-subject combinations. For open SQL, the Relational Database model allows for joins between tables but the queries quickly become complex (see examples in section 2, above) and for tables containing several hundred thousand rows, joins can be particularly time-consuming unless the multiple indexes in the closed system are applied. Tests 11 and 12 show that performing numerous individual queries rather than a single, more complex join operation can be more time-efficient, even if the end result is identical. However, this is programmatically more complex as it requires tailored scripts to be generated, and even then the SPARQL queries are generally quicker, particularly (as test 12 shows) for authors who have written a larger volume of papers.

Neither SPARQL nor SQL quick enough

This was a single, complex test which involved issues that lead to a struggle for both SPARQL (incomplete string querying) and open SQL (multi-table joins on large tables). Even the closed SQL system struggled with this: multiple indexing did not help with the 'like' query to the extent that it did with the other queries. In practice, the SemioViewer application breaks this query down into two stages: perform a search to get a paper ID (best performed using open SQL) and get the details of that paper and its authors (best performed using SPARQL). It is important to note that queries like this will exist when constructing applications for large-scale metadata stores, and the solution is to break it down into less complex queries and perform them sequentially using a suitable approach for each one. It is not possible to generalise at this stage as to whether a hybrid system is always the answer to large-scale metadata store querying, however given current technology and the typical scale of online digital libraries (several hundred thousand papers), the hybrid approach is currently the required solution within this domain.

4.6.3 Further Analysis

It is important to again point out that these experiments were performed on a single server using a particular instance of MySQL, on which the 3Store was built. The differences therefore can not be put down to superior hardware or database performance, but to the design differences between the RDBMS and RDF models of data representation. While it can be argued that the multiple indexing of the closed database provide better results for the SQL queries, this still leaves us with complex SQL statements performing JOIN operations on large tables, as well as the inherent problem of performing updates on what needs to be a live, frequently-updated system.

The results therefore show that in practice, the only realistic way for the SemioViewer application to work is to have both open SQL and SPARQL queries. While not typical Semantic Web applications, both the SemioViewer and the SPARQL-based web services and client pages described above require both SQL and

SPARQL queries in order to perform effectively, if they are to remain open to having regular data updates.

This is partly due to the design of SPARQL: certain features present in SQL are not included in SPARQL, such as there being no count(*) function and no 'like' facility within SPARQL. This is with good reason – for example, within SQL, 'like' is a syntactic term, usually denoting string similarity. In a SPARQL context, the question of whether a piece of data is 'like' another piece is more of a semantic question, and implies issues of contextual similarity of meaning. As such it is unlikely that 'like' would or indeed should be included in SPARQL as an equivalent to the SQL term. Therefore it remains the case that for the queries required in the experiments described in this chapter, a syntactic 'like' is required and as such SQL will continue to be necessary.

The requirement of having both SPARQL and SQL is also partly due to differing natures of SPARQL and SQL: as suggested above, SQL is better at 'identity' queries, SPARQL superior at 'relationship' queries. With metadata for ~700,000 papers and 9 million triples, the only practical approach when creating live, updateable 'semiometrics' applications is to use both: open SQL for initial searching and SPARQL for getting more in-depth data for each paper or author, including information needed for influence analysis.

4.7 Conclusion

While the statistics produced of the SemioViewer application are interesting in themselves, the main conclusion to be drawn in this chapter is that the RDF/SPARQL approach, along with a scalable triplestore solution, presents a viable alternative to SQL for large-scale metadata stores, particularly for queries based around relationship rather than identity. In this chapter examples have been shown from a working application where SPARQL out-performs open SQL on both the theoretical and empirical level, as well as examples of SQL out-performing SPARQL. It has also been shown that while a few simpler queries can be performed well using both approaches, there are very few that neither approach can handle in a reasonable time-frame: in these cases, simplifying queries provides the solution. In addition, it has been shown that for systems that do not require frequent updates, a closed, heavily-indexed is preferable as it requires only one data source (an SQL database) rather than both a database and RDF KB; however, it has also been shown that for large-scale metadata stores requiring frequent updating, a closed system is impractical. It has therefore been shown that when dealing with large-scale datasets featuring complex relationships and queries in the context, RDF and SPARQL can provide a dramatically improved performance over the conventional RDBMS/SQL approach for relationship-centric queries. Thus the second sub-hypothesis of this thesis has been proved: specifically, the query and update efficiency when using an ontology-based data store for importing and retrieval is better than when performing the same data browsing in a traditional SQL database.

While this chapter has dealt with the value of the ontological architecture of the semiometrics system, the following chapter focuses on the actual results produced by the system and empirically assesses the variety of semiometrics produced using three levels of data: paper, author and discipline.

Chapter 5 Practical Semiometrics

The previous chapter showed the value of an influence metrics system based around the principles of semantic web technologies, and the results of the experiment made clear the computational complexity and response time benefits of using an ontological approach to data storage. Having therefore shown the overall value of using ontologies and RDF to produce a viable data storage/retrieval paradigm, the question that next needs to be answered is whether the results produced by such a system are actually worthwhile. This chapter describes experiments performed using the semiometrics system and expert analysis to show both the inherent usefulness and flexibility of the ontology-based influence metrics system, improving on existing systems and thus proving the third sub-hypothesis of this thesis. This chapter also suggests which particular metrics are most useful when it comes to determining research influence.

The technical details of the system described in this chapter are covered in depth in Appendix B.

5.1 Problem Domain

The work described in this chapter builds on the data framework described in the previous chapters but addresses a specific question: whether the data produced by the system is worthwhile, and if it is worthwhile, which metrics (or combination of metrics) are most useful? The system is seeking to produce legitimate, useful statistics on research influence and therefore the problem tackled in this chapter is central to the worth of the entire system.

5.2 Overview of the Semiometrics System

Chapter four dealt with the data preparation processes and the overall architecture of the Semiometrics system, discussing the best query methods for particular data retrieval requirements. This leads to a final system design based on web client programs and server-side data storage, communicating through a web service structure that queries using both SPARQL and standard SQL approaches. See Figure B.2 in Appendix B for a detailed diagram of the overall system architecture.

5.3 Empirical Evaluation

The purpose of the experiments described in this chapter are to measure the effectiveness of the various influence metrics made possible by the semiometrics system. The evaluation compares a variety of weighted-metric approaches against straight citation counting for papers, and also applies the approach to the 'semantic zoom' level of authors. As part of these experiments, the different metrics are also compared against each other to determine optimum weightings for these various measures. A third level of 'semantic zoom' is also added for these experiments: the sub-discipline level. As part of the experiments it would be determined whether optimum weightings tend to be consistent across sub-disciplines of computer science, while dividing papers and people according to discipline allowed expert opinion in those specific fields to be elicited to determine the relative effectiveness of the techniques.

Two experiments are described in detail later in this chapter: the first deals with papers, the second with authors. In both cases, a number of different lists were created, drawing data only from papers in specific sub-disciplines, the lists comprising paper titles or author names ranked according to a variety of influence metrics such as citation count and authority rating. It is important to note at this stage that citations counted were citations from all papers in the corpus rather than just citations from papers within the sub-discipline, although the semiometrics system would allow for such metrics and future iterations of this experiment should take such metrics into account. Experts in each sub-discipline were also given a similar list of the top titles and names, and asked to rank those they were familiar with according to their relative influence on that specific field. The 'expert lists' and the various semiometric lists were then compared to determine the relative real-world usefulness of the semiometric measures being produced by the system, asking firstly whether they are more useful (both in terms of accuracy and practical usefulness) than traditional citation count alone (thus proving the third sub-hypothesis of this thesis) and secondly which of the metrics are the most effective and useful.

5.4 The Semiometrics Approach

As described in chapter four, there are two particular client systems that build on the SPARQL and SQL web services at the heart of the semiometrics system. The first client, a series of web pages forming an application, allows searching and browsing of ordered data. The second client, a semiometric viewer, allows users to browse individual papers and authors, returning their relative influence scores according to a variety of measures, and showing the chosen object alongside its graph-neighbours (citing documents for papers, co-authors for people). Both these clients run parallel to a Citeseer mirror and draw from Citeseer's dataset, which was augmented by the author disambiguation process described in chapter three to produce a populated ontology detailing the metadata held within Citeseer. Similarly, the dataset of the ACM digital library was also augmented and used to populate an ontology, stored in a 3Store on the same server as the Citeseer mirror and semiometrics web services.

5.4.1 Categorisation

While the two datasets (ACM and Citeseer) are quite similar in terms of metadata produced, one key difference is the pre-availability of document classification metadata in the ACM set. While services such as [ClassAKT] could be used to apply classifications to the Citeseer dataset, for the purposes of these experiments it was decided to use the categorisation provided by the ACM because (1) it was readily available and (2) it was applied by authors/editors of the original documents.

The ACM Computing Classification System is a standard computer science classification scheme for academic papers, comprising three coded levels (along with a fourth, uncoded, level based on specific subject descriptions), and each paper may contain one or more three-level coding. For example, the paper "Optimal agendas for multi-issue negotiation" (Fatima, Wooldridge, Jennings, 2003) is given the ACM code I.2.11, meaning the paper falls under the level 1 category I ('Computing Methodologies'), level 2 category 2 ('Artificial Intelligence') and level 3 category 11

('Distributed Artificial Systems'). Within this category there are four uncoded level 4 categories: 'Coherence and Coordination', 'Intelligent Agents', 'Language and Structures' and 'Multiagent Systems', however since these are uncoded categories, they are not taken into account by the ACM metadata element 'category code'. Therefore for the purposes of the experiments described in this chapter, only the top three levels of the ACM Classification System were used.

It is worth noting that papers are often given several classifications: for example, the paper described above is not only tagged with the I.2.11 category code but also with the code K.4.4 ("Computing Milieux", "Computers and Society", "Electronic Commerce"). While it is clear that the subject area of paper may indeed cover several of the topics contained in various parts of the ACM Classification System, it is also clear that the paper may apply more to one area than another. In many cases, those responsible for applying the classification may put the most central theme first in the list, however there is no way to tell the relative importance of those themes: if a paper has three category codes, there will be some cases where the first is much more important than the other two and others where the first two are equally important and the third less so. Beyond this observation, some papers seem to simply have their classifications listed in alphabetical order. For this reason, as is discussed in more depth in section 5.4.3, the significance of the ordering of category codes is not currently taken into account when performing grouping or analysis calculations: simply, if a paper is tagged with a particular category it may be used as part of the corpus of that category, no matter where that category is placed on the overall list for a given paper.

It is important to note at this stage that the AKT Reference Ontology, while containing a class relating to the research area of interest for people, does not cover the notion of papers having a particular research topic associated with them. For this reason, as is covered in more depth in Appendix B, one of the extensions made to the AKT Reference Ontology allowed for the application of ACM categories to individual papers.

5.4.2 Metrics

A number of outputs are produced by the two clients. Some of these are non-metric results, such as the graphical view of paper citations and co-author networks, however the majority of the outputs are metrics, either giving scores for individual papers/authors or producing ranked lists to show the relative influence of the paper/author in the context of other similarly-influential work. The specific metrics produced vary according to the level of granularity being considered (paper, author, discipline) but are all based around various ways of using citation information to determine influence. Chapter two contains a discussion of the relative merits of pure citation counting and more qualitative approaches: what is presented below is simply a description of the various metrics made available by the system, allowing the user to choose or weight them as appropriate.

Paper Count

As stated in chapter two, if we define 'impact' or 'influence' in terms of scientific usage, then simply counting the papers produced by a particular author, institution or discipline does not necessarily clearly reflect the influence of that producer. However, as a simple measure, the sheer number of refereed papers produced does reflect at least partially the influence of the person or group concerned. That said, citation measures such as the mean citation count and the H-Index take the number of papers authored into account and as they are citation-based metrics, they may be considered genuine influence metrics. The paper count metric is, however, made available for users to view and use as they see fit for granularity levels higher than that of individual papers.

Citation Count

This is available for the granularity levels of authors and papers. This is the simplest genuine 'influence' metric and the one most commonly used in the literature, forming the basis for the Journal Impact Factor, as described in chapter 2 and in [Garfield 1994]. Although, as stated in chapter 2, there are many criticisms of using pure citation counting, it still forms the basis of other measures such as authority, betweenness centrality and H-Index: with zero citations, these measures would also be zero. Therefore the simple citation count metric, particularly for papers, is one of the key metrics produced by the system. At levels higher than individual papers, it is the summation of all citation counts of all papers in a particular grouping, for instance all papers authored by a particular author.

Mean Citation Count

This is a compositional influence metric available at granularity levels higher than single papers. Most commonly associated with authors, this is an improvement on the straight citation count as it also takes into account the number of papers authored. However, it is not clear that it should replace the summation citation count as a measure of influence altogether: as stated above, while the number of papers authored is important, it is also the case that a total citation count does reflect the complete number of references given to authors by their peers. Therefore for higher levels, both total citation count and mean citation count are presented as metrics.

HITS Authority Rating

Kleinberg's hubs-and-authorities algorithm [Kleinberg 1998], as described in chapter two, provides two potential measures ('hubs' and 'authorities'), as identified by [Shadbolt *et al.* 2006]. Of these, the authority metric is the most useful as it represents papers that have been cited by hubs. The model predicts that certain papers (typically literature reviews) tend to cite the most influential papers in a given field. Although both hub and authority scores must be calculated alongside each other, the semiometrics system only outputs the authority figure as a measure of influence: while it would be an accomplishment to write a paper that is highly cited by literature reviews alongside other influential papers (ie a high authority paper) it would be less influential to write a paper that cites many important papers but may itself never be cited (ie a high hub paper). Therefore only authority scores are a legitimate influence metric and as such as output by the system. It is clear that 'authority', both conceptually and as defined by Kleinberg, is central to the broad notion of 'influence' as used throughout this thesis and its use in the experiments described in this chapter reflect this. This metric is available at all granularity levels and is normalised between 0.0 and 1.0.

Compositional Authority Metrics

As with the pure citation counts, compositional authority metrics are produced by the system for the levels above individual paper. Total authority scores are produced, as are mean authority scores, and as well as being available for viewing on their own, they are used along with citation counts to produce weighted combination metrics.

Weighted Combination

While authority scores show the relative importance of the citations received by a paper, it remains the case that citation count is still of some value. The question is: how important is the authority score compared to the citation count? The system by default produces a 1:1 weighted ratio between them. Specifically this means that, for a paper, each citation (worth 1.0) is added to the authority of that citation (between 0.0 and 1.0) allowing authoritative citations to be worth up to twice as much as pure citation values. Other weightings can be given to the system, favouring citation count or authority, but the straight 1:1 ratio is the default. This is available at all levels: at levels higher than individual papers, the combination scores for the individual papers are calculated and the mean calculated. The graphical visualiser application, which shows (by size of node) the relative influence of papers or authors along with their closely-related nodes, uses this measure (weighted 1:1) by default to determine node size.

H-Index

As described in chapter two, Hirsch's H-Index [Hirsh 2005] is an attempt to measure not just the individual influence of a paper, but to calculate the influence of a scientist over a career. Specifically, the H-index takes into account the number of papers written and the number of citations received by individual papers. Various criticisms and improvements on the H-Index have been proposed since Hirsch first published the algorithm: however the fact remains that the H-Index is sufficiently interesting and simple enough to calculate that it is clearly of use as a potential influence metric at the author level. Future iterations of the semiometrics system would allow experiments to be performed to determine whether there is value in applying H-Index calculations to granularity levels higher than author.

Modified H-Index

Some of the modifications proposed to the H-Index are covered in chapter two. These include Egghe's G-Index to account for very highly cited papers, and Bjelobrk and Zuckerman's proposal for dividing the H-Index by the number of years of active scientific service. However, a novel modification to the H-Index metric has been devised in line with the work described in this thesis as one of the output metrics. One of the key problems with the H-Index is that, like the pure citation count, it is a citation-based metric that treats all citations as equal. By introducing the element of authority-based relative citation importance, it is possible to produce a modified H-Index that takes account of the fact that some citations are more important than others.

In the context of the semiometrics system, the modified H-Index is based on the weighted combination with ratio 1:1 as described above. The normal H-Index is calculated as the number of papers which have higher than *h* citations, where *h* is the total number of papers written by that author. The modified H-Index is calculated as the number of papers that have a higher weighted combination than *h*, where the weighted combination allows each citation value (default 1.0) to be as much as doubled depending on its authority value. Therefore each modified H-Index score will be at least as much as the existing H-Index score, but the idea is that if an author has written particularly influential (authoritative) papers that have slightly fewer citations than the *h* threshold, the addition of the authority data will boost the citation count to take it over the threshold.

One problem with both the H-Index and the modified H-Index is that the data sources are currently incomplete in terms of citations: the Citeseer and ACM datasets only include citations from documents within each corpus. While this will improve as the datasets become more common and integrated, the values currently fall some way short of the H-Index scores predicted by Hirsh. However, it has been included as there is currently great interest in the H-Index and also because, as the datasets represent a reasonably unbiased cross-section of computer science, the relative ranked list positions are expected to closely mirror what they would be if full citation information was made available.

5.4.3 Avoided Metrics

For a variety of reasons, a number of potential metrics are not produced by the system at present. This is because the metrics are considered unreliable either because they are conceptually unproven or because the results produced are clearly incorrect. A summary of the key metrics not produced by the system is given below.

Betweenness Centrality

Betweenness Centrality is, as stated in chapter two, a metric that shows how many trans-graph pathways pass through a given node. This is particularly interesting when a high proportion of pathways on a citation graph pass through a particular node, as it implies that that node is a turning point in the discipline concerned, a theory shown to be true in small (sub-100 size) graphs by [Chen 2004] in the CiteSpace visualiser. However, in larger graphs such as the citation networks in the ACM and Citeseer datasets, calculating the Betweenness Centrality of each node in the graph does not yield useful results: in particular, it appears that the datasets contain quite a number of individual, unconnected citation graphs, some much larger than others. The problem is that less influential papers end up in small graphs, and thus potentially gain a large Betweenness Centrality score. Therefore, while the Betweenness Centrality score (between 0.0 and 1.0) for such a node is correct in relation to other node scores from that graph, it does not relate in any way to the scores in other graphs. As a result, Betweenness Centrality scores, which were calculated, like the Authority scores, using the JUNG framework applied over the entire dataset, are not useful as absolute influence scores: they are always relative to the particular graph of which they are a part. To become useful, the individual graphs need to be identified and given some kind of weighting relating to their relative importance to the overall dataset, which is a nontrivial task even at the conceptual level. Chen's use of Betweenness Centrality, conversely, works because the influence scores are only relative to the individual graph, and that is all that CiteSpace attempts to show. So until a relative graph weighting algorithm can be successfully defined and introduced to the semiometrics system, the Betweenness Centrality algorithm will not be included in the system.

Chronological-based Metrics

While [Sombatsompop *et al.* 2004] showed that over-time citation histograms follow a half-life decay model, there remains controversy over the usefulness or otherwise of metrics that use time-based criteria. Specifically, different systems currently in existence make different assumptions about the time-gap between the original paper being created and the citing paper being written. For example, the Rexa system [McCallum *et al.* 2006] implements a half-life value model to citations, where an older median citation age is translated as meaning a less influential paper, author or topic. Similarly, the original Journal Impact Factor calculation as described in chapter two only takes into account citations from the preceding two years, ignoring older citations.

Conversely, however, discussions with experts in other fields including humanities and social sciences have yielded the opposite response: in these fields classic papers are often highly cited for years after their creation, and the longer the citation lag or half-life, the higher the influence of a given paper. At face value this appears to suggest a difference according to discipline, however it is sufficiently unclear how to interpret citations over different time spans that no measure was implemented in the current version of the semiometrics system.

Order of Author List

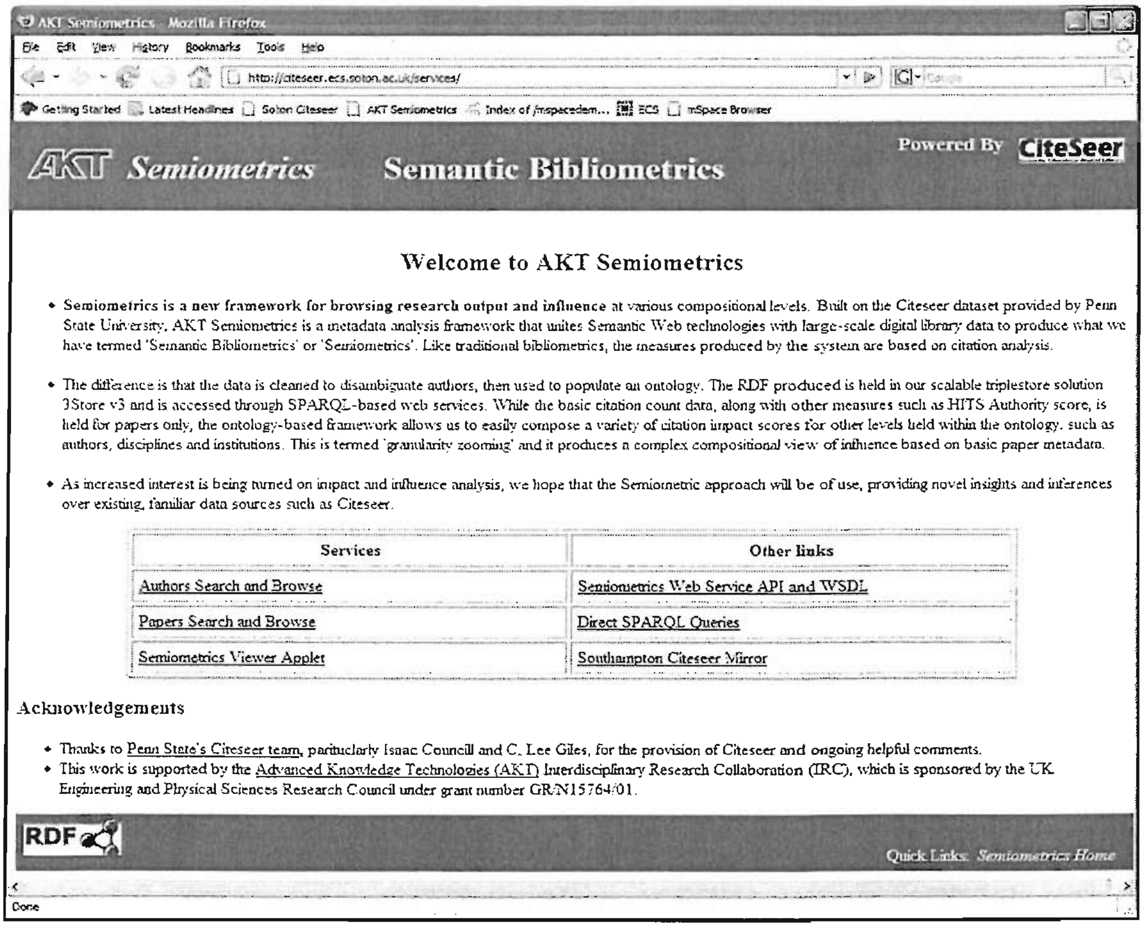
While it is standard practise to list paper authors according to the amount of the paper they are responsible for, there is no way to tell the proportion of responsibility each author has. For example, it may be the case that a paper with three authors has one author primarily responsible for the content, while another paper with three authors has two equally-responsible primary authors. In both cases, a list of three authors will be given for the paper, in the order of responsibility, but this will not take into account the differences in responsibility of the second author in each list. While it is clear that if at all possible, more responsible authors should be given a higher score for high-influence papers, it is not clear that this will be possible given the existing system. For this reason, the semiometrics system currently treats all authors as equally important.

Order of ACM Category Code

Similar to the authors list, the ACM categories for a paper are usually listed in order of relevance. As described in section 5.4.1, the ordering of categories, like the ordering of authors, does not show the relative importance of those values. For this reason, the semiometrics system currently considers all papers that contain a given category code as fully belonging to that category.

5.4.4 Interface

The iteration of the semiometrics system used to perform the experiments described in this chapter is entirely web-based. The web services architecture, which include both SQL and SPARQL data querying structures, feeds into the PHP-based web client ordered lists, as described in chapter four, while the semiometric viewer application is re-created in Java Applet form.



Welcome to AKT Semiometrics

- **Semiometrics is a new framework for browsing research output and influence** at various compositional levels. Built on the Citeseer dataset provided by Penn State University, AKT Semiometrics is a metadata analysis framework that unites Semantic Web technologies with large-scale digital library data to produce what we have termed 'Semantic Bibliometrics' or 'Semiometrics'. Like traditional bibliometrics, the measures produced by the system are based on citation analysis.

- The difference is that the data is cleaned to disambiguate authors, then used to populate an ontology. The RDF produced is held in our scalable triplestore solution 3Store v3 and is accessed through SPARQL-based web services. While the basic citation count data, along with other measures such as HITS Authority score, is held for papers only, the ontology-based framework allows us to easily compose a variety of citation impact scores for other levels held within the ontology, such as authors, disciplines and institutions. This is termed 'granularity zooming' and it produces a complex compositional view of influence based on basic paper metadata.

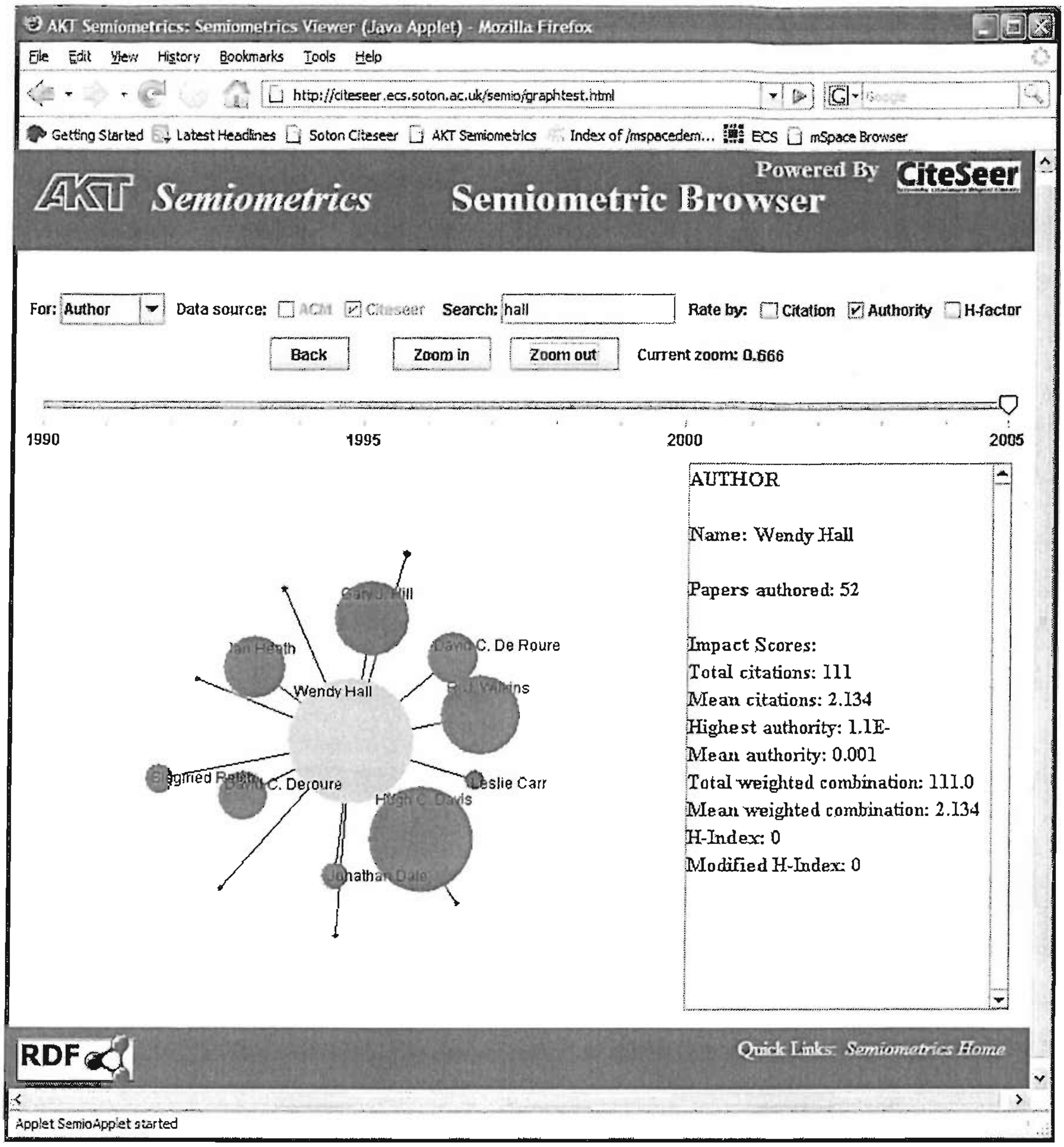
- As increased interest is being turned on impact and influence analysis, we hope that the Semiometric approach will be of use, providing novel insights and inferences over existing, familiar data sources such as Citeseer.

Services	Other links
Authors Search and Browse	Semiometrics Web Service API and WSDL
Papers Search and Browse	Direct SPARQL Queries
Semiometrics Viewer Applet	Southampton Citeseer Mirror

Acknowledgements

- Thanks to Penn State's Citeseer team, parituclarly Isaac Councill and C. Lee Giles, for the provision of Citeseer and ongoing helpful comments.
- This work is supported by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01.

Figure 5.1 : The AKT Semiometrics Website



AKT Semiometrics: Semiometrics Viewer (Java Applet) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://citeseer.ecs.soton.ac.uk/semio/graphtest.html
Getting Started Latest Headlines Soton Citeseer AKT Semiometrics Index of /mspacedem... ECS mSpace Browser
AKT Semiometrics
Semiometric Browser
Powered By CiteSeer
Back
Zoom in
Zoom out
1990
1995
2000
2005
Gary J. Hill
Ian Heath
David C. De Roure
Wendy Hall
R. J. Watkins
Siegfried Reich
David C. DeRoure
Leslie Carr
Hugh C. Davis
Jonathan Dale
RDF
Quick Links: Semiometrics Home
Applet SemioApplet started

For: Author

Data source: □ ACM ☑ Citeseer

Search: hall

Rate by: □ Citation ☑ Authority □ H-factor

Current zoom: 0.666

AUTHOR

Name: Wendy Hall

Papers authored: 52

Impact Scores:
Total citations: 111
Mean citations: 2.134
Highest authority: 1.1E-
Mean authority: 0.001
Total weighted combination: 111.0
Mean weighted combination: 2.134
H-Index: 0
Modified H-Index: 0

Figure 5.2 : Semiometrics Viewer Applet

5.5 Experiment

In order to test the effectiveness of the semiometric system, and thus prove the third sub-hypothesis of this thesis, the specific metrics set out in section 5.4.2 above were tested according to expert analysis. The experiments described in this section were performed using categorised ACM data and were performed on paper and author data. Experts from two specific fields were asked to perform ordering on the same papers and author data and the results were compared with the lists produced by the various system metrics in order to find the most effective combination of potential metrics, as well as demonstrating the effectiveness of the semiometrics framework in a practical setting.

5.5.1 Limitations and Expectations

While these experiments are important in determining the usefulness or otherwise of the metrics described in section 5.4.2, it is important to note that the third sub-hypothesis of this thesis is concerned with proving that a set of metrics other than pure citation counting yields a measure of influence more closely related to that provided by expert analysis. For the prototype system used for this experiment, this means that a successful outcome should not be that the actual metrics produced are the final, absolute figures, but instead that the trend should be for ranked lists to match more closely those of the experts.

For example, the data used in all the experiments described in this thesis were taken from two sources: Citeseer and the ACM Digital Library. A paper within the ACM Digital Library will have a certain number of citations – however, these citations are only from other documents contained within the ACM Digital Library, and so the citation count will obviously be incomplete. Similarly, Citeseer's data is also incomplete although this is not due to it being restricted to one particular set of published papers: in Citeseer's case the problem is simply one of lack of coverage. While it contains some 700,000 papers and has calculated citation links between many of these to a high degree of accuracy, it remains incomplete and, along with other online DLs covering Computer Science, it comfortably contains less than half the current literature of computer science [Petricek *et al.* 2005]. Additionally, while it may initially appear that Citeseer should give more of a cross-section of computer science as it is not restricted to just one publisher (as is the case with the ACM dataset), in actual fact Citeseer tends to gather its papers from single sites it finds through crawling or recommendation. As a result of this, Citeseer will often scrape papers from academic web pages of individuals containing complete career bibliographies, and thus end up with a fairly complete collection for that author, while containing others purely by virtue of their co-authoring with an authors whose website Citeseer has scraped, or indeed missing them out altogether. Citeseer's data is therefore skewed in favour of particular authors much as the ACM's dataset is skewed in favour of those authors who tend to publish their results via the ACM.

Therefore, due to the nature of the data being worked with, the absolute results given by the system are to be considered less important than the relative positions within ordered lists. In particular, citation count and direct citation-related measures such as the H-Index score are not accurate representations due to the incomplete data at present. While online systems exist for performing H-Index calculations such as [Schwartzbach's H-Number Calculator], via sources such as Google Scholar, it remains the case that single-source bibliometric c alculations will always fall short. However, provided the distribution of data is not too skewed, citation counts should be fairly representative samples of the total number and therefore ranked lists become a legitimate tool. In the case of both Citeseer and the ACM data, while they are skewed as described above, they are not particularly skewed in favour of any particular research area, and as the next section describes, that is the focus of these experiments. However, it remains important to note the current limitations of the system in terms of absolute results, particularly of more purely citation-based metrics. As the number of data sources increases and the percentage coverage of a given subject domain grows increasingly complete, the absolute figures will become more exact – however the value of the system lies in relative ranked positions within lists and it is this that the semiometrics system is, and will continue to be, primarily outputting.

5.5.2 Data and details

For the purposes of these experiments, as set out in section 5.3, it was decided to focus on specific subsets of computer science. While Citeseer's data does not usually contain 'subject area' metadata (a subset of 12,197 documents have been classified but this represents only about 1.7% of Citeseer's total document set), the ACM Digital Library classifies each paper against its own breakdown of computer science, as described in section 5.4.1. Using this classification for each paper, and by implication each author, it became possible to perform a variety of calculations using the system to produce a series of ordered rankings of papers and authors in those fields. Experts in those fields would then be consulted and their 'expert rankings' compared against those produced by the system for the various metrics. This would produce two effects in particular. Firstly, a correlation between the expert lists and any of the system-produced lists would back up the value of the system-based results. Secondly, comparisons between the relative effectiveness of each of the system-produced lists would allow determination of which approaches are more effective, and which less so, and indeed whether there is any difference across different domains.

Specifically, the domains chosen were Information Search and Retrieval (ACM category H.3.3) and Distributed Artificial Intelligence (ACM category I.2.11), due to the large number of documents held in the ACM corpus for each category (11143 for

H.3.3 and 5079 for I.2.11) and the availability of highly-rated experts in those fields. The metrics chosen were aligned with those listed in 5.4.2, specifically: for papers, citation count and weighted combination of citation count and authority score; for authors, total citation count, mean citation count, total weighted combination, mean weighted combination, H-Index and the new modified H-Index, as described in section 5.4.2. The experiments were performed using the semiometrics framework as described in chapter 4 and the following section summarises and discusses the results produced.

In terms of measuring similarity between the expert-produced ordered lists of results and those produced by the system, a list-similarity metric was used. A variation on the bubble-sort algorithm (as defined in the 'Definitions and Abbreviations Used' section) was employed: specifically, the minimum number of bubble-sort changes required to turn the one list into the other was measured and expressed as a proportion of the maximum number of possible changes for the worst possible list. For example, if the expert-produced list contained five items {1, 2, 3, 4, 5} and the system-produced list ranked these instead as {1, 4, 2, 3, 5}, it would take two changes to change the second list into the first: firstly swapping the 4 and the 2 and secondly swapping the 4 and the 3. The maximum number of changes possible in a set of five results would be if the list had come out backwards, ie {5, 4, 3, 2, 1}: such a result would require 10 changes. Therefore the {1, 4, 2, 3, 5} list requires 2 out of a possible 10 changes, which show a difference of 0.2 or a similarity of 0.8. This method allows lists of varying lengths to be normalised between 0.0 and 1.0. In the results section that follows, list-similarity rather than list-difference is given as the result, but list-difference is always (1.0 – list-similarity). In terms of results, it is not clear what should be regarded as a 'good' or 'bad' result, but any similarity score over 0.5 would show that a list is more similar than not. A score of 1.0 would reflect two identical lists and a score of 0.0 would show two perfectly dissimilar lists. A random selection of results would, on average, produce a similarity of 0.5 and thus any result over 0.5 may be seen to reflect two lists which are more similar that the statistical average.

5.5.3 Results

The ordered lists produced by the system are to be compared with 'expert' ranked lists of the same papers and people. The first set of results given should therefore be the results provided by experts in the two fields. In the cases of both Information Search and Retrieval and Distributed Artificial Intelligence, these results have been provided by expert groups in their fields and neither they themselves nor their papers were ranked as part of the test. Specifically, two experts from each field contributed the ranked lists, amalgamating their two views into a single list for each field. These lists were then used in these experiments. At the request of the some of the experts their names, groups and institutions will not be mentioned. The expert rankings for papers and authors for each category were as follows (reading columns downwards first rather than rows left-to-right):

Information Search and Retrieval (ACM Category H.3.3) – Papers
1. The anatomy of a large-scale hypertextual web search engine (Brin, Page, 1998).
2. A language modeling approach to information retrieval (Ponte, Croft, 1998).
3. Authoritative sources in a hyperlinked environment (Kleinberg, 1999).
4. Scatter/Gather: a cluster-based approach to browsing large document collections (Cutting, Karger, Pederson, Tukey, 1992).
5. Automatic resource compilation by analyzing hyperlink structure and associated text (Chakrabarti, Dom, Raghavan, Rajagopalan, Gibson, Kleinberg, 1998).
6. Query expansion using local and global document analysis (Xu, Croft, 1996).
7. Focused crawling: a new approach to topic-specific web resource discovery (Chakrabarti, van den Berg, Dom, 1999).
8. KMS: a distributed hypermedia system for managing knowledge in organisations (Akscyn, McCracken, Yoder, 1998).

Information Search and Retrieval (ACM Category H.3.3) – Authors		
1. Gerard Salton	9. Marti A. Hearst	17. James F. Allen
2. W. Bruce Croft	10. Prabhakar Raghavan	18. Andreas Paepcke
3. Donna Harman	11. Soumen Chakrabarti	19. Jan O. Pedersen
4. Sergey Brin	12. Amanda Spink	20. Philip S. Yu
5. Lawrence Page	13. Norbert Fuhr	21. Krishna Bharat
6. Chris Buckley	14. Oren Etzioni	22. Byron Dom
7. Jon Kleinberg	15. Monika R. Henzinger	23. David Gibson
8. James P. Callan	16. Justin Zobel	

Distributed Artificial Intelligence (ACM Category I.2.11) - Papers
1. Distributed rational decision making (Sandholm, 1999).
2. Agents that buy and sell (Maes, Guttman, Moukas, 1999).
3. Collaborative interface agents (Lashkari, Metral, Maes, 1994).
4. Seven good reasons for mobile agents (Lange, Oshima, 1999).
5. Collaborative plans for complex group action (Grosz, Kraus, 1996).
6. The Michigan Internet AuctionBot (Wurman, Wellman, Walsh, 1998).
7. The dynamics of reinforcement learning in cooperative multiagent systems (Claus, Boutilier, 1998).
8. The interdisciplinary study of coordination (Malone, Crowston, 1994).
9. Multiagent reinforcement learning (Hu, Wellman, 1998).
10. Coalitions among computationally bounded agents (Sandholm, Lesser, 1997).
11. Planning and acting in partially observable stochastic domains (Kaelbling, Littman, Cassandra, 1998).
12. Learning collaborative information filters (Billsus, Pazzani, 1998).
13. WebMate: a personal agent for browsing and searching (Chen, Sycara, 1998).
14. A hierarchical approach to wrapper induction (Muslea, Minton, Knoblock, 1999).

Distributed Artificial Intelligence (ACM Category I.2.11) – Authors		
1. Victor Lesser	10. Munindar P. Singh	19. Paolo Ciancarini
2. Pattie Maes	11. Maria Gini	20. William E. Walsh
3. Michael P. Wellman	12. Danny B. Lange	21. Robert H. Guttman
4. Katia Sycara	13. Makoto Yokoo	22. Robert Tolksdorf
5. Michael Wooldridge	14. Onn Shehory	23. Michael J. Pazzani
6. Tuomas Sandholm	15. Thomas W. Malone	24. Bamshad Mobasher
7. Edmund H. Durfee	16. Peter R. Wurman	25. Alexandros G. Moukas
8. Sarit Kraus	17. Kevin Crowston	
9. Barbara J. Grosz	18. Franco Zambonelli	

These results were compared against ordered lists produced using the semiometrics system to see firstly whether there was significant correlation between the results and secondly to see which metrics produced the most accurate correlations. The complete set of ordered lists output by the system per subject area and metric method are summarised for each result using the notation described in section 5.5.2 – for example, {1, 4, 2, 3, 5} would show a list where the item fourth in the expert list was then ranked second in the system-produced list. Additionally, the list-similarity scores are given when the system-produced results are compared against the above lists, showing how similar the system-produced lists are to those ranked by experts.

List-edit distance results for papers

H.3.3 **Citation Count**	**H.3.3** **Weighted Combination**
1. The anatomy of a large-scale hypertextual web search engine (Brin, Page, 1998).	1. The anatomy of a large-scale hypertextual web search engine (Brin, Page, 1998).
8. KMS: a distributed hypermedia system for managing knowledge in organisations (Akscyn, McCracken, Yoder, 1998).	8. KMS: a distributed hypermedia system for managing knowledge in organisations (Akscyn, McCracken, Yoder, 1998).
2. A language modeling approach to information retrieval (Ponte, Croft, 1998).	2. A language modeling approach to information retrieval (Ponte, Croft, 1998).
3. Authoritative sources in a hyperlinked environment (Kleinberg, 1999).	3. Authoritative sources in a hyperlinked environment (Kleinberg, 1999).
5. Automatic resource compilation by analyzing hyperlink structure and associated text (Chakrabarti, Dom, Raghavan, Rajagopalan, Gibson, Kleinberg, 1998).	5. Automatic resource compilation by analyzing hyperlink structure and associated text (Chakrabarti, Dom, Raghavan, Rajagopalan, Gibson, Kleinberg, 1998).
6. Query expansion using local and global document analysis (Xu, Croft, 1996).	7. Focused crawling: a new approach to topic-specific web resource discovery (Chakrabarti, van den Berg, Dom,

	1999).
7. Focused crawling: a new approach to topic-specific web resource discovery (Chakrabarti, van den Berg, Dom, 1999).	6. Query expansion using local and global document analysis (Xu, Croft, 1996).
4. Scatter/Gather: a cluster-based approach to browsing large document collections (Cutting, Karger, Pederson, Tukey, 1992).	4. Scatter/Gather: a cluster-based approach to browsing large document collections (Cutting, Karger, Pederson, Tukey, 1992).
Final order: {1, 8, 2, 3, 5, 6, 7, 4}	**Final order: {1, 8, 2, 3, 5, 7, 6, 4}**
List-similarity: 0.679	**List-similarity: 0.643**

I.2.11 Citation Count	I.2.11 Weighted Combination
8. The interdisciplinary study of coordination (Malone, Crowston, 1994).	8. The interdisciplinary study of coordination (Malone, Crowston, 1994).
2. Agents that buy and sell (Maes, Guttman, Moukas, 1999).	2. Agents that buy and sell (Maes, Guttman, Moukas, 1999).
9. Multiagent reinforcement learning (Hu, Wellman, 1998).	9. Multiagent reinforcement learning (Hu, Wellman, 1998).
5. Collaborative plans for complex group action (Grosz, Kraus, 1996).	7. The dynamics of reinforcement learning in cooperative multiagent systems (Claus, Boutilier, 1998).
3. Collaborative interface agents (Lashkari, Metral, Maes, 1994).	4. Seven good reasons for mobile agents (Lange, Oshima, 1999).
4. Seven good reasons for mobile agents (Lange, Oshima, 1999).	5. Collaborative plans for complex group action (Grosz, Kraus, 1996).
14. A hierarchical approach to wrapper induction (Muslea, Minton, Knoblock, 1999).	14. A hierarchical approach to wrapper induction (Muslea, Minton, Knoblock, 1999).
1. Distributed rational decision making (Sandholm, 1999).	1. Distributed rational decision making (Sandholm, 1999).
11. Planning and acting in partially	11. Planning and acting in partially

observable stochastic domains (Kaelbling, Littman, Cassandra, 1998).	observable stochastic domains (Kaelbling, Littman, Cassandra, 1998).
12. Learning collaborative information filters (Billsus, Pazzani, 1998).	13. WebMate: a personal agent for browsing and searching (Chen, Sycara, 1998).
13. WebMate: a personal agent for browsing and searching (Chen, Sycara, 1998).	12. Learning collaborative information filters (Billsus, Pazzani, 1998).
7. The dynamics of reinforcement learning in cooperative multiagent systems (Claus, Boutilier, 1998).	3. Collaborative interface agents (Lashkari, Metral, Maes, 1994).
10. Coalitions among computationally bounded agents (Sandholm, Lesser, 1997).	10. Coalitions among computationally bounded agents (Sandholm, Lesser, 1997).
6. The Michigan Internet AuctionBot (Wurman, Wellman, Walsh, 1998).	6. The Michigan Internet AuctionBot (Wurman, Wellman, Walsh, 1998).
Final order: {8, 2, 9, 5, 3, 4, 14, 1, 11, 12, 13, 7, 10, 6}	**Final order: {8, 2, 9, 7, 4, 5, 14, 1, 11, 13, 12, 3, 10, 6}**
List-similarity: 0.571	**List-similarity: 0.549**

Figure 5.3 : List-edit distance results for papers

The results, reflecting the relative list similarities for papers as produced by the experts and the system, show a correlation between the lists produced by the system and the expert lists. In particular, the first research area (H.3.3) shows a system-produced citation-based list that is identical to the expert-produced list with the exception of two papers: firstly the paper ranked eighth by the experts ('KMS: a distributed hypermedia system for managing knowledge in organisations', Akscyn, McCracken, Yoder, 1998) which is more highly cited than predicted by the experts, and secondly the fourth paper ('Scatter/Gather: a cluster-based approach to browsing large document collections', Cutting, Karger, Pederson, Tukey, 1992) which is cited less. The list-similarity score of 0.679 can therefore be considered relatively high as there are only those two anomalous papers in the entire list, albeit quite out-of-position. The combination-based list, where citation count and authority measures are combined, shows little difference to the citation only list, with the exception of the seventh paper

('Focused crawling: a new approach to topic-specific web resource discovery', Chakrabarti, van den Berg, Dom, 1999) being ranked one place higher, due to its citations being from more hub-like papers such as literature reviews. Again, the list-similarity score of 0.643 appears a relatively good score.

The second research area (I.2.11) shows a less clear correlation between the two lists, although the pattern of results show that papers ranked higher by the experts tend to appear in the early sections of the lists, and lower ones later. Of the differences, one paper in particular ('The interdisciplinary study o f coordination', Malone, Crowston, 1994 – eighth on the expert list) ranks much more highly than predicted by the experts due to it being highly cited but not directly related to the 'distributed artificial intelligence' area – instead, the authors have specified it as being more related to a sub-area of I.2.11, "Coherence and coordination", which is less directly relevant to the expert group who provided the list. However, the list-similarity scores again show the lists to be quite similar – 0.571 and 0.549 show that there are some statistical similarities between the system-produced lists and those produced by the experts.

In both cases, it is interesting to note that although there is in all cases a correlation between the system-produced and expert-produced lists, the citation-only list is marginally more accurate than the list based on a combination of citations and authority scores. Although only slightly difference, it does show the flexibility of the semiometrics system in producing a variety of metrics and allowing more user control.

List-edit distance results for authors

H.3.3: Total Citations		
11. Soumen Chakrabarti	19. Jan O. Pedersen	20. Philip S. Yu
1. Gerard Salton	4. Sergey Brin	22. Byron Dom
7. Jon Kleinberg	5. Lawrence Page	13. Norbert Fuhr
2. W. Bruce Croft	17. James F. Allen	15. Monika R. Henzinger
3. Donna Harman	21. Krishna Bharat	23. David Gibson
8. James P. Callan	12. Amanda Spink	6. Chris Buckley
9. Marti A. Hearst	14. Oren Etzioni	10. Prabhakar Raghavan
16. Justin Zobel	18. Andreas Paepcke	
Final order: **{11, 1, 7, 2, 3, 8, 9, 16, 19, 4, 5, 17, 21, 12, 14, 18, 20, 22, 13, 15, 23, 6, 10}**		
List-similarity: 0.605		

H.3.3: Mean Citations	
9. Marti A. Hearst	5. Lawrence Page
3. Donna Harman	10. Prabhakar Raghavan
6. Chris Buckley	7. Jon Kleinberg
8. James P. Callan	1. Gerard Salton
4. Sergey Brin	2. W. Bruce Croft
Final order: {9, 3, 6, 8, 4, 5, 10, 7, 1, 2}	
List-similarity: 0.378*	

H.3.3: Total Weighted Combination		
11. Soumen Chakrabarti	20. Philip S. Yu	15. Monika R. Henzinger
1. Gerard Salton	4. Sergey Brin	22. Byron Dom
7. Jon Kleinberg	5. Lawrence Page	12. Amanda Spink
2. W. Bruce Croft	18. Andreas Paepcke	14. Oren Etzioni
3. Donna Harman	21. Krishna Bharat	23. David Gibson
8. James P. Callan	13. Norbert Fuhr	6. Chris Buckley
9. Marti A. Hearst	16. Justin Zobel	10. Prabhakar Raghavan
17. James F. Allen	19. Jan O. Pedersen	
Final order: **{11, 1, 7, 2, 3, 8, 9, 17, 20, 4, 5, 18, 21, 13, 16, 19, 15, 22, 12, 14, 23, 6, 10}**		
List-similarity: 0.577		

H.3.3: Mean Weighted Combination	
9. Marti A. Hearst	5. Lawrence Page
3. Donna Harman	10. Prabhakar Raghavan
6. Chris Buckley	7. Jon Kleinberg
8. James P. Callan	1. Gerard Salton
4. Sergey Brin	2. W. Bruce Croft
Final order: {9, 3, 6, 8, 4, 5, 10, 7, 1, 2}	
List-similarity: 0.378*	

H.3.3: H-Index		
4. Sergey Brin	6. Chris Buckley	18. Andreas Paepcke
1. Gerard Salton	2. W. Bruce Croft	13. Norbert Fuhr
15. Monika R. Henzinger	9. Marti A. Hearst	23. David Gibson
20. Philip S. Yu	17. James F. Allen	3. Donna Harman
19. Jan O. Pedersen	12. Amanda Spink	21. Krishna Bharat
7. Jon Kleinberg	5. Lawrence Page	16. Justin Zobel
8. James P. Callan	14. Oren Etzioni	22. Byron Dom
10. Prabhakar Raghavan	11. Soumen Chakrabarti	
Final order: **{4, 1, 15, 20, 19, 7, 8, 10, 6, 2, 9, 17, 12, 5, 14, 11, 18, 13, 23, 3, 21, 16, 22}**		
List-similarity: 0.656		

H.3.3: Modified H-Index		
4. Sergey Brin	6. Chris Buckley	16. Justin Zobel
1. Gerard Salton	2. W. Bruce Croft	13. Norbert Fuhr
15. Monika R. Henzinger	9. Marti A. Hearst	23. David Gibson
20. Philip S. Yu	18. Andreas Paepcke	3. Donna Harman
19. Jan O. Pedersen	12. Amanda Spink	21. Krishna Bharat
7. Jon Kleinberg	5. Lawrence Page	17. James F. Allen
8. James P. Callan	14. Oren Etzioni	22. Byron Dom
10. Prabhakar Raghavan	11. Soumen Chakrabarti	
Final order: **{4, 1, 15, 20, 19, 7, 8, 10, 6, 2, 9, 18, 12, 5, 14, 11, 16, 13, 23, 3, 21, 17, 22}**		
List-similarity: 0.652		

I.2.11: Total Citations		
6. Tuomas Sandholm	9. Barbara J. Grosz	17. Kevin Crowston
4. Katia Sycara	16. Peter R. Wurman	21. Robert H. Guttman
5. Michael Wooldridge	15. Thomas W. Malone	19. Paolo Ciancarini
3. Michael P. Wellman	25. Alexandros G. Moukas	22. Robert Tolksdorf
8. Sarit Kraus	14. Onn Shehory	7. Edmund H. Durfee
1. Victor Lesser	11. Maria Gini	24. Bamshad Mobasher
13. Makoto Yokoo	12. Danny B. Lange	18. Franco Zambonelli
2. Pattie Maes	10. Munindar P. Singh	
20. William E. Walsh	23. Michael J. Pazzani	
Final order: **{6, 4, 5, 3, 8, 1, 13, 2, 20, 9, 16, 15, 25, 14, 11, 12, 10, 23, 17, 21, 19, 22, 7, 24, 18}**		
List-similarity: 0.663		

I.2.11: Mean Citations	
6. Tuomas Sandholm	13. Makoto Yokoo
1. Victor Lesser	8. Sarit Kraus
5. Michael Wooldridge	9. Barbara J. Grosz
7. Edmund H. Durfee	3. Michael P. Wellman
12. Danny B. Lange	15. Thomas W. Malone
11. Maria Gini	4. Katia Sycara
14. Onn Shehory	2. Pattie Maes
10. Munindar P. Singh	
Final order: {6, 1, 5, 7, 12, 11, 14, 10, 13, 8, 9, 3, 15, 4, 2}	
List-similarity: 0.514*	

I.2.11: Total Weighted Combination		
7. Edmund H. Durfee	10. Munindar P. Singh	20. William E. Walsh
3. Michael P. Wellman	12. Danny B. Lange	21. Robert H. Guttman
6. Tuomas Sandholm	17. Kevin Crowston	15. Thomas W. Malone
5. Michael Wooldridge	24. Bamshad Mobasher	23. Michael J. Pazzani
11. Maria Gini	19. Paolo Ciancarini	2. Pattie Maes
1. Victor Lesser	9. Barbara J. Grosz	18. Franco Zambonelli
16. Peter R. Wurman	13. Makoto Yokoo	14. Onn Shehory
4. Katia Sycara	8. Sarit Kraus	
22. Robert Tolksdorf	25. Alexandros G. Moukas	
Final order: **{7, 3, 6, 5, 11, 1, 16, 4, 22, 10, 12, 17, 24, 19, 9, 13, 8, 25, 20, 21, 15, 23, 2, 18, 14}**		
List-similarity: 0.647		

I.2.11: Mean Weighted Combination	
6. Tuomas Sandholm	13. Makoto Yokoo
2. Pattie Maes	8. Sarit Kraus
5. Michael Wooldridge	9. Barbara J. Grosz
7. Edmund H. Durfee	3. Michael P. Wellman
12. Danny B. Lange	15. Thomas W. Malone
11. Maria Gini	4. Katia Sycara
14. Onn Shehory	1. Victor Lesser
10. Munindar P. Singh	
Final order: {6, 2, 5, 7, 12, 11, 14, 10, 13, 8, 9, 3, 15, 4, 1}	
List-similarity: 0.505*	

<table>
<tr><th colspan="3">I.2.11: H-Index</th></tr>
<tr><td>2. Pattie Maes</td><td>9. Barbara J. Grosz</td><td>15. Thomas W. Malone</td></tr>
<tr><td>3. Michael P. Wellman</td><td>19. Paolo Ciancarini</td><td>23. Michael J. Pazzani</td></tr>
<tr><td>7. Edmund H. Durfee</td><td>16. Peter R. Wurman</td><td>24. Bamshad Mobasher</td></tr>
<tr><td>4. Katia Sycara</td><td>12. Danny B. Lange</td><td>20. William E. Walsh</td></tr>
<tr><td>1. Victor Lesser</td><td>17. Kevin Crowston</td><td>6. Tuomas Sandholm</td></tr>
<tr><td>5. Michael Wooldridge</td><td>11. Maria Gini</td><td>14. Onn Shehory</td></tr>
<tr><td>8. Sarit Kraus</td><td>21. Robert H. Guttman</td><td>25. Alexandros G. Moukas</td></tr>
<tr><td>10. Munindar P. Singh</td><td>22. Robert Tolksdorf</td><td></td></tr>
<tr><td>18. Franco Zambonelli</td><td>13. Makoto Yokoo</td><td></td></tr>
<tr><td colspan="3"><b>Final order:</b>
<b>{2, 3, 7, 4, 1, 5, 8, 10, 18, 9, 19, 16, 12, 17, 11, 21, 22, 13, 15, 23, 24, 20, 6, 14, 25}</b></td></tr>
<tr><td colspan="3"><b>List-similarity: 0.790</b></td></tr>
</table>

<table>
<tr><th colspan="3">I.2.11: Modified H-Index</th></tr>
<tr><td>2. Pattie Maes</td><td>9. Barbara J. Grosz</td><td>15. Thomas W. Malone</td></tr>
<tr><td>3. Michael P. Wellman</td><td>16. Peter R. Wurman</td><td>23. Michael J. Pazzani</td></tr>
<tr><td>7. Edmund H. Durfee</td><td>17. Kevin Crowston</td><td>24. Bamshad Mobasher</td></tr>
<tr><td>4. Katia Sycara</td><td>12. Danny B. Lange</td><td>20. William E. Walsh</td></tr>
<tr><td>1. Victor Lesser</td><td>18. Franco Zambonelli</td><td>6. Tuomas Sandholm</td></tr>
<tr><td>5. Michael Wooldridge</td><td>11. Maria Gini</td><td>13. Makoto Yokoo</td></tr>
<tr><td>8. Sarit Kraus</td><td>21. Robert H. Guttman</td><td>25. Alexandros G. Moukas</td></tr>
<tr><td>10. Munindar P. Singh</td><td>22. Robert Tolksdorf</td><td></td></tr>
<tr><td>19. Paolo Ciancarini</td><td>14. Onn Shehory</td><td></td></tr>
<tr><td colspan="3"><b>Final order:</b>
<b>{2, 3, 7, 4, 1, 5, 8, 10, 19, 9, 16, 17, 12, 18, 11, 21, 22, 14, 15, 23, 24, 20, 6, 13, 25}</b></td></tr>
<tr><td colspan="3"><b>List-similarity: 0.790</b></td></tr>
</table>

** Asterisk indicates incomplete lists and scores produced by the system, meaning the actual list-similarity scores are probably substantially lower than those presented here and therefore these should not be used as indicators of research influence.*

Figure 5.4 : List-edit distance results for authors

These author-based results are very encouraging in terms of both showing similarity measures that are useful and those that are not useful. Firstly, it quickly appears that, for the author statistics, *mean* citations and *mean* weighted combination scores have little correlation to the expert-produced lists. Indeed, the statistics produced by the system for these lists rank a number of the people on the original lists in such a lowly position that it wasn't possible to locate them within the parameters of the system (which produced the top 100 for each section). While it would be possible to locate them by extending the parameters, the semiometrics system was not extended in this way simply because the results even as they stood were poor, especially in the case of the H.3.3 dataset, and with other results being outside the top 100 they can barely be seen to match the top 25 as ranked by the experts.

The reason for the poor performance of the mean number of citations and weighted combination is unclear, but it does appear that authors tend to write a good number of papers, many of which are not highly cited simply because they represent the reporting of the ongoing work of an individual or research group rather than any particular breakthrough which will be highly influential and as such, highly cited. Additionally, papers with a high 'hub' score tend not to be particularly highly cited, and these also bring down the mean scores. For these reasons, total citations, total combination and the two H-Index measures are more suitable measures of research influence as they take more account of highly-cited papers while giving less importance to low-cited progress or literature review papers.

Having therefore removed the 'mean' scores, there are four measures left to determine author influence. In all cases the lists produced by the system are more similar than different, and in all but one case (H.3.3 total weighted combination) the similarity scores were over 0.6. For both H.3.3 and I.2.11, the total weighted combination list was slightly less accurate than the total citation count list, although in reality the two lists are always fairly close to each other: for example, in the H.3.3 list, James F. Allen rises from 20th to 15th in the rankings when moving from considering total citations to total weighted combination, pushing down by one those between on the total citations list. Apart from that move, there is only one other minor switch in position as Monika R. Henzinger drops two places from 14th to 16th, resulting in a couple of other authors moving up by one position each. The I.2.11 list features more changes from the total citation count list to the total weighted combination list, but overall the changes more or less balance themselves out, the weighted combination list being only slightly less accurate to the expert list than the citation count list. For example, while Onn Shehory (14th on the expert list) drops from 14th on the total citation count list to 19th on the weighted combination list, Maria Gini (11th on the expert list) climbs from 16th on the citation list to 12th on the weighted combination list. Overall, the two measures seem to produce similar results, which is encouraging given that the authority score (determined by citation graph analysis) is likely to be a little inaccurate in an incomplete citation graph such as the ACM dataset: as more data is added (using processes such as that described in chapter six of this thesis) it is clear that a more accurate set of authority scores can and will be produced over time. However, it seems that even with the existing data it is acceptable to use the weighted combination of citation count and authority score as a similarity metric, alongside the citation count score alone. It is a little disappointing to report that the weighted combination list-similarity tends to be slightly lower than the citation count list-similarity alone, however as stated above, the lack of a complete citation graph means these results, even as they stand, are encouraging as they produce results clearly quite similar to those produced by the experts.

However, it is equally clear that the two H-Index lists are even more similar to those produced by the experts. For H.3.3 and even more so for I.2.11, the H-Index scores produce lists that tend to very closely match the view of the experts, particularly at the highest level: the top ten results of the H.3.3 H-Index and modified H-Index contain only three results from outside the 'expert' top ten; the top ten results of the I.2.11 H-Index and modified H-Index is missing just one result from the corresponding 'expert' top ten (Barbara J. Grosz, 9th on the expert list). The overall H-Index results are excellent and bear closer resemblance to the expert lists than the other scores. This represents an affirmation of Hirsch's assertion that the H-Index is a superior measure of research influence than total or mean citation measures alone; the fact of the availability of this data through the semiometrics system and its placement alongside the other results, allowing user choice as to which metric or metrics to use, shows the effectiveness of the semiometrics system as an all-encompassing set of services, a central point for determining research influence, which is of more use than disparate and unconnected sources such as Citeseer's 'most cited author' page [Citeseer Author Statistics] and [Schwartzbach's H-Index Calculator], which is based on the Google Scholar dataset.

A couple of interesting anomalous results do stand out and bear special mention from the sets of author results. Firstly, from the I.2.11 results, the author ranked ninth on the expert list (Barbara J. Grosz) appears lower on all the four valid result lists. Harvard University's Professor Grosz, an expert in agent-based computing who has developed a highly-influential theory of discourse structure, has authored a number of highly-influential papers but overall has fewer total papers authored than other scientists on the list. It is therefore reasonable to understand her ranking being higher by the experts than by the semiometrics system, which is more based on statistics than professional reputation. From this anomalous result we can conclude that future iterations of the semiometrics system should take more account of extremely highly-cited papers, giving additional rankings to authors who may have written fewer overall papers, but whose papers may be very influential indeed. Incorporation into the system of a measure such as the G-Index, as described in section 2.1.4, might provide a solution to this problem as it gives additional value to exceptionally highly-cited papers.

Secondly, and also from the I.2.11 set, the author ranked twenty-third on the expert list (Michael J. Pazzani) is consistently ranked much higher by the semiometrics system than expected: in particular when considering the weighted combination list, in which he appears second. UCI's Professor Pazzani is indeed a widely-published academic and researcher; however his specific expertise lies in the field of machine learning algorithms, an area which is generally listed under section I.2.6 of the ACM classification rather than I.2.11. However, the breadth of Professor Pazzani's work allows sufficient encroachment into the I.2.11 area that the statistics regarding his papers in I.2.11 (many of which are also listed under I.2.6) are actually very high and comparable with leaders from that specific field. However, experts in I.2.11 would naturally tend to consider Pazzani more a 'machine learning' person than a 'distributed artificial intelligence' person and thus rank him lower on the I.2.11 list. Two particular conclusions can be drawn from this. Firstly, that a possible improvement to the system might be to introduce a weighting to the ACM classifications attributed to a paper, for example giving more importance to papers that list (for example) I.2.11 as the first category and less to papers that list I.2.11 lower down the list. This would require a change to the ontology and the database schema to allow for relative positions of these categories to be taken into account. Secondly, however, it should be noted that this anomaly shows that perhaps in certain circumstances, the system can in fact aid and augment expert analysis, since it can show that even though a particular category may not be the primary area of a research for a given scientist, they may still be far more active and influential in that area than the experts give them credit for, based on existing reputation.

5.6 Comparisons

The results given above can be compared to some degree against other existing systems and metrics to show their effectiveness, despite the shortcomings of these other methods as described in chapter 2.

As a primary example, the Citeseer team at Penn State University periodically produce a list of the 10000 top-cited authors according to the Citeseer database [Citeseer Author Statistics], and it is possible to look up on this list the names ranked by the experts and determine their relative positions on this list. There are two particular shortcomings with this approach. Firstly, the citations counted refer to all the papers produced by these people, rather than just those within a particular category of computer science, since Citeseer does not hold classification metadata for papers. Secondly, Citeseer does not contain disambiguated author names within its standard database. Instead it performs simple string-matching disambiguation when producing this list, leading to inaccurate results. For example, this approach combines the scores for James F. Allen (17th on the expert-ranked list for H.3.3) with all other researchers named J. Allen, thus bringing his name to the top of the list by some distance. Conversely, W. Bruce Croft (2nd on the expert-ranked H.3.3 list), who appears in the Citeseer list as both B. Croft and W. Croft, has his citation total split as a result: B. Croft drops to 5th on the Citeseer-based results, W. Croft appearing 22nd. Overall, however, the results based on this Citeseer list are good: the H.3.3 results show a list-similarity of 0.632 with the expert-produced list, and the I.2.11 results showing an impressive 0.790 list-similarity to the view of the experts (matching the 0.790 score of the Semiometrics system when calculating H-Index scores for I.2.11). These two results are an improvement over the citation count results as given by the semiometrics system, which is not surprising as the ACM dataset only takes into account citations from other ACM-published documents, and therefore is less accurate than the wider document base of Citeseer. However, the results are not so different as to render the ones produced by the Semiometrics system useless: indeed, since the Citeseer data is also held in the Semiometrics system, such results could be obtained simply by choosing Citeseer as the data source rather than ACM.

Another widely-used measure, the H-Index, can be calculated over Google Scholar using [Schwartzbach's H-Index Calculator] as described in section 5.5 above. Feeding the names from the two lists into Schwartzbach's tool produced scores almost identical to the Semiometrics system for H-Index scores: list-similarities of 0.668 for H.3.3 and 0.797 for I.2.11. As this system uses Google Scholar as its data source, the results would again be expected to be higher than those produced by the Semiometrics system as more citations are going to be taken into account than simply those from ACM-published sources.

However, the problem with using the Citeseer list and the H-Index calculator in this way is that while their results can be compared with the Semiometrics system results, they are incapable of producing such a list of authors or papers in the first place for sub-disciplines. The Semiometrics system, holding data in ontology format, is capable of showing such cross-section author (and paper) results based on ACM research area of papers, and these results can then be given to experts to perform rankings which can in turn be used in experiments such as those described above. The Citeseer list or Schwartzbach's H-Index calculator are not capable of producing sub-discipline level results in the first place, and this is a key development within the Semiometrics system.

5.7 Conclusions

The Semiometrics system is therefore shown to be an advancement upon existing systems and a tool of value as it not only allows calculation of a variety of metrics in one place, including citation counting and H-Index scores for authors, but it is also capable, due to the ontological nature of its data format, of easily producing results for sub-disciplines for authors and papers given only a simple categorisation of papers, in this case the ACM classification. These lists can then be used by other tools and systems as required, particularly via the publicly-available web services aspect of the system architecture. However, the results in this section also clearly show the results from the system, incomplete as it currently is in terms of data, are comparable with other widely-used metrics such as the Citeseer list or the H-Index calculator. Since these results are also now available in one place, from one system and data store, it is clear that the Semiometrics system provides several key advances over existing systems: the results from the system are useful (ie list-similarity always >0.5 and consistently >0.6), comparable with existing systems, available in one place and capable of being viewed at sub-discipline level. Therefore the system can clearly be seen to be a useful contribution in terms of practical influence metric usage, and thus the third sub-hypothesis of this thesis is proved.

The following chapter looks at a further advantage of using an ontology-based approach to mediate data for research influence analysis. Building on the framework and successful results already described in this thesis, the extensibility of data and ease of integration when using an ontology is leveraged to merge the Citeseer and ACM datasets into one usable, de-duplicated collection – an exemplar of an ongoing process required by the fourth sub-hypothesis: the need for simple, live data integration.

Chapter 6 Metadata Integration

As has been shown in the preceding chapters, a viable system based on ontology-format data allowing semantic bibliometrics is not only possible but advantageous over traditional RDBMS-based approaches. Two primary reasons have been shown in the preceding chapters as evidence for the superiority of the ontology approach: the suitability of the RDF approach to query efficiency given the need for an open system, and the effectiveness of the output, able to compose multiple layers of metrics (paper, author, sub-discipline) purely from paper citation information. However, given the need for the Semiometrics system to be a viable real-world application, the system must also take advantage of one additional key feature of an ontology-based data format: the ability to easily add information from heterogeneous sources. This chapter describes the additional elements required to be added to the data preparation process in order to apply the semiometric system across multiple datasets. It concludes that using ontologies and specifically the reasoning capabilities provided by the Web Ontology Language [McGuiness & Van Harmelen 2004] provides a simpler, superior data fusion capability for open-ended sources, thus proving the fourth sub-hypothesis.

The technical implementation details of the system described in this chapter are given in Appendix B.

6.1 Problem Domain

The work described in this chapter addresses the problem of creating a knowledge base that accesses data from more than one major data source. The requirement is firstly for a system that can run automatically, matching data from (to begin with) two disparate sources, and secondly for the mappings between those two datasets to be sufficiently meaningful that the joint dataset be useful in real-world scenarios. Specifically, the two data sources chosen were the metadata sets of Citeseer and the ACM Digital Library, and the real-world scenarios to be used are the two semiometric client applications as described in chapters four and five.

6.2 Overview of system

The data preparation process for the existing semiometric system as described in the previous chapters details the steps required to prepare data for usage in the semiometrics system. One of the first considerations to be made for the merger of two or more sets of data is to determine a point in the process at which the merging could best be performed. The experiments described in this chapter have three aims: firstly to determine the best point in the existing data preparation process, secondly to determine the d-precision and d-recall (novel metrics introduced in section 3.4) scores for merging the two datasets in order to determine the best balance of algorithms and thirdly to determine whether the data thus created is, in fact, useful in the existing system.

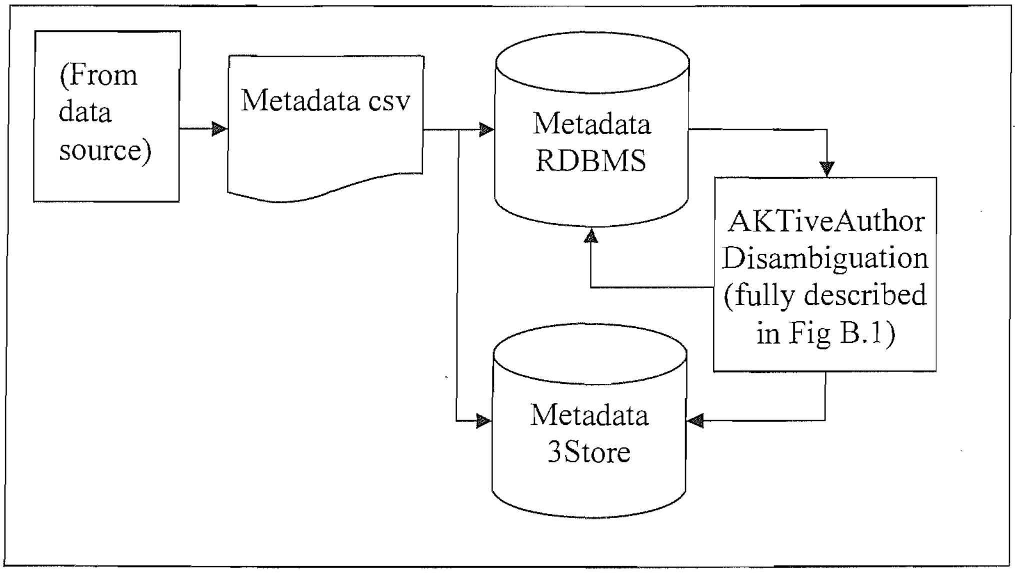


Figure 6.1 : Overview of system as described in previous chapters

Figure 6.1 shows the existing data preparation process as described in previous chapters. From this it can be seen that the data exists in four formats at various stages of the process: initial harvested data (usually some kind of XML format such as OAI:PMH), csv flat-file data, RDBMS tables and the RDF/ontology version as stored in the 3Store. Of these, the first can be discounted as a possible point at which mergers can be performed: for a standard merger script to be created, the data must be in a standard format and while certain standards such as OAI:PMH are becoming increasingly common, it remains the case that some datasets, including the ACM Digital Library metadata set, are held in a proprietary XML format, which would require pre-processing before a merging script could be applied. As we already have pre-processing as part of the existing data preparation, it would not make any sense to duplicate the effort and pre-process the data into a different format: therefore the first potential point of the existing process at which we could attempt data merging is when the two datasets are held in a standard csv format. The remaining two formats in which data is held (RDBMS and RDF triplestore) are both legitimate potential points at which merging could be applied, so this gives a total of three data formats to be used as potential merger points: data held in csv file, data held in relational database in standard schema, and data held in a triplestore asserted against a standard ontology.

6.3 Empirical Evaluation

The experiments described in this chapter have as their central aim to create a standard merger script or module that could become a part of the overall data preparation process. The datasets to be used in the experiments were two static, known datasets already harvested and held in the various formats: snapshots of the Citeseer database from mid-2004 and the ACM Digital Library from mid-2005. One particular advantage of using these two datasets was that a comparison could be made with results from the Citeseer team's attempts at discovering the degree of overlap between the Citeseer and ACM sets. [Citeseer/ACM]

The experiments, as stated above, had three aims: the discovery of the best data format for merging, the d-precision and d-recall metrics for the merged data and the real-world usefulness of this data. In practise, this means creating three systems (one for each data format) and comparing the metrics and usefulness figures for each of these.

6.3.1 Chapter 4 Revisited

Learning from the previous process as described in chapter 4, it is clear that it is not just papers that should be matched, but it would also be highly advantageous to match authors as well. This allows us to take the author matching process of multiple-author papers out of the client programs and keep them as quick and simple as described in chapter 4. It also will help match across a few of the outliers described in chapter 3, increasing the d-recall figures, however as with chapter 3 it is important that d-precision is very high, even if this means a loss of d-recall. These observations are expanded upon in the following sections.

It is also important at this point to note the differences between both the aims and methodology described in chapter 3 and the aims and methodology described regarding the work described in this chapter. In chapter 3, the aim is to disambiguate document authors primarily using graph-based techniques. As the following sections describe, the aim of the experiments described in this chapter is to disambiguate both papers and authors using string-matching algorithms.

6.3.2 In-depth View of Matching Process

One of the key advantages of asserting data against an ontology is the ease with which new data can be added to the knowledge base, extending the ontology as necessary. In the case of the AKT Reference Ontology, already extended as described in chapter 4 and detailed in Appendix B, well-structured classes dealing with papers and authors are already in place. For representing the mapping of data across datasets, two steps are required: firstly the asserting of all the existing data in a single knowledge base and secondly the creation of a mapping file, which is then asserted in the same knowledge base.

Within the Web Ontology Language (OWL), a mapping construct data instances exists: triples with the *owl:sameAs* predicate can be asserted and this relationship can be used to perform suitable inference calculations. For example, a URI representing a Citeseer paper could be linked to a URI representing the same paper in the ACM corpus using *owl:sameAs* as the predicate, thus allowing citations and other metadata held by just one of the papers to become common to both, thereby yielding more accurate influence scores (such as citation count). For the purposes of mapping between the two datasets, *owl:sameAs* statements were to be created and asserted in the RDF triplestore. While it would be possible to produce this same information in RDBMS format, probably involving a new table that allowed for mappings between paper and author IDs, the RDBMS system does not automatically allow for *owl:sameAs* style inference statements to be created, and these would have to be dealt with programmatically. Additionally, since most of the complex queries in the semiometric client programs are performed on triplestore data, having the mapping data in the triplestore would therefore provide useful data for the existing system and would also provide flexibility of being able to easily add new information when the datasets are updated or a new dataset added.

The aim therefore is to find a method that will take data from one of three possible sources (csv file, RDBMS, triplestore) and output a set of *owl:sameAs* statements mapping instances between the two sets of papers and authors. Other classes, such as citing papers, paper subject classifications and author affiliations can be inferred from the mapped first-class objects.

6.3.3 Paper Matching

The key question in this process is to determine what is and isn't a match. To help with this, it was decided to perform initial matches on papers using title similarity metrics, therefore identifying candidate matches, and then to perform author matching on the authors of those papers, to confirm or deny the candidate matches. This would mean that, in most cases, the author names being compared and matched would be quite distinct as it is unusual for authors with very similar names to author the same paper. While authors with similar names do occasionally co-author papers, and examples of this are given below, the overall situation is such that string similarity metrics can be performed on the author names knowing that (1) the paper in question has already been matched, so we should be able to find a mapping between two authors, and (2) because there are a small number of potential mappings that need to be tested in each case, and on the whole they are going to be quite different, the similarity threshold that needs to be met in order to declare a match may be quite low. Papers that produce a match higher than the chosen similarity threshold for more than one author would simply have neither of the matches committed. While this would potentially lower the d-recall metric for successful matching, the possibility would still be there for other papers to perform the same match on the same author.

In terms of matching papers, it was decided in the prototype stage to perform the matching using paper titles that met three conditions (expanded upon in more detail below): firstly, that the titles of the paper in the two databases matched exactly; secondly, that the title string was seventeen characters or more in length (see below for a full explanation of why seventeen was chosen); and thirdly, that the title string contained more than one word. It is worth noting that while a potential fourth condition was considered – that of only matching papers with the same year of publication – it was decided not to implement this condition as, particularly in the Citeseer database, many papers had badly-parsed or non-existent year fields. This leaves three key conditions, all of which deal with string similarity metrics. Each of the criteria requires a brief explanation.

Exact title matching

While mis-typed or incorrectly parsed titles would be problematic in this case, the actual number of papers with incorrect titles in the two datasets used in the experiment was very small. It was therefore determined that for the prototype experiments, exact title matching would be used, and if the d-recall figure was deemed unacceptable, string metric calculations would be used instead. However, because of the importance normally placed on the paper title both by users entering data manually, and because of the pre-eminence of the title on a typical first page being automatically parsed, it was hoped that the d-recall figures would mean that further refining would not be required.

Title string seventeen characters or more in length

The principle here is straightforward: papers entitled, for example, 'Software Agents', tend to be re-occurring titles so having a minimum length threshold would be help in removing the commonly re-occurring titles, which tend to be shorter. However, it is also the case that mis-entered or mis-parsed titles tend to be short in length: incorrect titles such as those which are either blank altogether or having titles such as 'Research Report' or 'Position Paper' would be picked up by including a minimum limit on characters. The limit was varied during the prototype process and eventually set at a minimum of seventeen for reasons set out in the following section.

Title string contains more than one word

During the prototyping process, it was useful to have this constraint in place as the principle was clear that single-word titles would almost always be potentially ambiguous. Whether they refer to a system name (such as "CryptoManager++") or a generic subject area (such as "Pseudorandomness"), this group also included common incorrect titles such as 'Acknowledgements' and 'Bibliography'. Rather than put together a 'stop-word' style list of unacceptable titles, it was observed that all such titles fell into the category of not containing more than one word, and as such this was implemented. Programmatically, this was expressed as a boolean value determined by whether or not the string contained the ASCII space character. After the completion of the prototyping process, it was determined that the longest single-worded matching paper title across the two databases (each containing several hundred thousand paper titles) in question was the title 'Pseudorandomness', sixteen letters in length. The threshold for the minimum length of title (as described in the previous section) was therefore set to be seventeen allowing, in theory, this third condition to be dropped. However, for use in future dataset matching, this condition was retained in the program although not used in the experiments described in this chapter.

6.3.4 Author Matching

While the processes of both manual entering and automated parsing of paper titles results in a high degree of correct titles, the same is not true of author name processing. This is partly because automated parsing systems are less successful with author name recognition, which tend to be in smaller type and less consistently placed within a page, and partly because of inconsistent name types. The latter of these is the more dominant problem: as described in chapter three, one of the major problems with author disambiguation is the variety of standard ways to refer to the same person: 'N. Jennings' 'Nicholas R. Jennings' and 'Prof. Nick Jennings' all refer to the same real-world scientist. However, unlike the method described in chapter three we are not using a graph-based approach to author disambiguation: in this case we are simply trying to match a very small number of authors of a single paper.

The matching process is therefore different and considerably simpler: instead of attempting to match metadata elements associated with authors, we simply need to find the equivalent name in each paper. Often these will be identical: automatic parsing of an author name in both datasets for an individual paper should yield the same name variant in both cases. For instance, if the paper in the first dataset refers to the author as N. R. Jennings, the second dataset should also refer to the author as N. R. Jennings as both systems are parsing the exact same paper, which refers to N. R. Jennings. However, manual entry of data can lead to variants emerging, as can inconsistent parsing (particular of smaller print), so some account needs to be taken of potential variances in names.

While the family-name-based approach as used by AKTiveAuthor could be applied here, the problem domain is slightly different: in the AKTiveAuthor case we are looking to perform initial clustering of data, and therefore are looking for a large group of potential matches which will be refined; in this case we are looking to perform the actual matching process. Therefore it was decided to implement this section using string similarity metrics applied across the author name.

A variety of string similarity measures are available, as discussed in chapter two. In particular, the Levenshtein string-edit distance [Levenshtein 1965] was considered useful in this case as it would allow for both baldy-parsed names and variances in references to be measured. Given the small number of potential matches within each paper (the Citeseer dataset contains a mean of 2.304 authors per paper and the ACM dataset a mean of 2.116 authors per paper), the similarity threshold was, as stated above, set to be relatively low. In implementation terms, it was decided to use the SimMetrics package [Chapman 2004] which has a number of string similarity measures, all of which are normalised to produce results between 0 and 1 (including Levenshtein), which would allow design and prototyping changes to be made if alternative measures were to be used instead of or in conjunction with the basic Levenshtein edit-distance calculations.

6.3.5 Prototyping: three potential match-points

Three prototype scripts were created, each one taking data from the three potential match-points identified above. In each case, the prototype attempted to match papers only, as a proof-of-concept that the system would be capable of producing useful results in a timely fashion. As covered in the implementation details (Appendix B), the systems in this section were programmed in Java. While a lot of the data handling described in this thesis were implemented in Perl, the use of the SimMetrics package (available only in Java and C#/.NET) meant that either the program would be created in Java or would require use of Perl's Inline::Java function. For simplicity, it was decided to implement the matching program entirely in Java.

The first attempt was to match using csv data. This is the simplest form of a flat data file, from which both the RDBMS and RDF data are derived. In order to describe this section and the conclusions drawn from it, it is necessary to cover some of the technical details more fully set out in Appendix B. While throughout the work described in this thesis, database input has been handled by a Java program and RDF creation from .csv source by Perl, the process has been identical: the terms are read in from a .csv file line-by-line into an array, then processed and read out into the correct format. In order to perform the process required in this chapter, it is necessary to use Java (for the use of SimMetrics) and it is required to read in at least one set of titles entirely into memory for testing against the other set. This process is both time-consuming and memory-heavy: indeed during the prototyping phase, several OutOfMemoryExceptions were encountered and the Java Virtual Machine heap size had to be reset to a higher maximum value. While eventually a working prototype was created, it was clear that the system was far from perfect and sufficiently slow to show that, if at all possible, one of the other two methods would be preferable.

The second attempt was made using SQL queries across the two databases. MySQL allows for cross-database select queries to be made. Specifically, for the prototype system, matching of papers was performed using the following query:

```
SELECT DISTINCT t1.Title, t1.articleID, t2,articleID
FROM acm.articles as t1, oai_citeseer.articles as t2
WHERE t1.Title = t2.Title AND LENGTH(t1.Title) > 16;
```

The results of this query were written out line by line to an ontology. In experiments paralleling those described in chapter four, two types of database were used: firstly one with minimal indexing allowing frequent updates, secondly one with indexing on all key table columns. As with chapter four, the prototype results showed the size of the tables made 'JOIN' operations impractically slow on the minimally-indexed database; however the fully-indexed databases responded to the query in less than a second. Whereas the experiments described in chapter four found the triplestore approach to be superior because of its open-ended nature and its need for live data updates, the process being considered here is determined with making further RDF data, rather than being part of the semiometric querying process itself. Thus the need for frequent updates and live data is less pressing: this process is by its nature a background batch update process, so allowances can be made for the slower insert and update statements that result from using a heavily indexed database. The SQL query approach was therefore considered a viable possibility.

The third attempt was made using SPARQL queries on an RDF knowledge base which already contained both datasets. Specifically, for the prototype system, the following query was attempted:

```
PREFIX akt: <http://www.aktors.org/ontology/portal#>
PREFIX spt: <http://www.aktors.org/ontology/support#>

SELECT distinct ?t ?p ?p2
WHERE
  {
    ?p2 akt:has-title ?t .
    ?p akt:has-title ?t .
  }
```

This is a simple-looking query: it asks the knowledge base for the URIs and titles of all documents that have been described using the *has-title* relation, in order that further processing could be used to determine matching titles and thus candidate equivalent URIs. However, the real-time complexity of this query – specifically, the need to search all triples in both datasets that contain the has-title query (of which there were over a million) before merging the two groups on matching titles – meant that the system was not able to return a reasonable set of matches in a reasonable time frame. Indeed, mirroring some of the results from chapter four, it was clear that the nature of this query was one based around identity rather than relationship: specifically, this query was searching for matching titles by considering all titles from all sources rather than looking for triples with particular predicate-subject combinations. It should also be noted that while the above query takes into account the requirement to match titles, the minimum length and minimum number of words requirements are not implemented. These could be implemented using a regular expression filter on the above query, but this would raise the query time still further so was not implemented at this prototype stage. The SPARQL approach was therefore considered unsuitable for usage in this system.

Therefore it was decided that the system would be best implemented using queries to a standard heavily-indexed RDBMS. The remaining two parts of the experiment in this chapter – determining the d-precision and d-recall given the algorithms chosen, and the overall usefulness of the data – were performed by extending the RDBMS-based prototype to include a query that covered both paper and author data, writing out a suitable RDF file containing *owl:sameAs* statements and importing it into the knowledge base. Specifically, matching of papers and authors was performed using the following query:

```
SELECT DISTINCT t1.articleID, t2.articleID, t1.Title, t3.acmID,
  t3.Author, t4.authorID, t4.canonicalName
FROM acm.articles as t1, oai_citeseer.articles as t2,
  acm.authors as t3, oai_citeseer.canon as t4
WHERE t1.Title = t2.Title AND LENGTH(t1.Title) > 16
   AND t1.articleID = t3.ArticleAuthored
   AND t2.articleID = t4.documentID
```

This query is similar to the one used by the prototype except for the additional inclusion of the author names and author IDs for each record. Including author information increases the number of tuples returned by the query as instead of one tuple being returned per matched paper, one tuple per author is returned. However, programmatically this allows for the author names to be tested against one another using the SimMetrics implementation of Levenshtein string-edit distance as described above. The *owl:sameAs* structure allowed the program to write out IDs of matching instances of authors independently of the papers with which they are associated. It was noted that duplicate author matches may appear in numerous papers, so a vector containing matched author IDs was also populated as the *owl:sameAs* statements were being written to file, and this vector was checked before each *owl:sameAs* statement was written to make sure the match had not been found before, which would lead to redundancy and inefficient use of the 3Store.

The system was programmed and run over the full sets of Citeseer and ACM data, and took approximately 2 minutes to complete. However, before discussing the results, it is important to consider the metrics used to determine the success or failure of the experiments.

6.3.6 Metrics

Of the three aims of the experiments described in these chapters, one had already been answered: merging data from an RDBMS source was shown to be superior to merging from csv or triplestore sources (and indeed was the only practical way of performing the merge). This left two aims: determining the success of the matching algorithm and the overall usefulness of the data thus created.

D-precision/D-recall

The first of these is similar to the aims of the author disambiguation work described in chapter three, and utilise the same novel metrics as introduced in section 3.4. Specifically, we are considering an information retrieval exercise and therefore there will be two types of results: d-precision and d-recall. D-precision in this case refers to the set of matched papers or authors: how many of those matched were correctly matched? D-recall, conversely, refers to the set of authors that should have been matched: how many of that set were, in fact, correctly matched, and how many were left out? As we are considering a different matching algorithm for papers and authors, we will therefore have four results – d-precision and d-recall for both papers and authors.

As was briefly mentioned above, one of the lessons learned from chapter three was the importance of high d-precision when matching authors, even if this is at the expense of d-recall. The nature of the author disambiguation problem is that false negatives are a serious problem: if two David B. Johnsons are joined, along with all their papers, the statistics will refer to a person who doesn't exist and neither of the two real David B. Johnsons will have correct statistics. However, if they are separate entities within the system, even if some of their papers are not attributed to them (ie lower d-recall), the statistics still have some meaning. In the case of merging datasets, lower d-recall figures are made more acceptable by the fact that other matched papers elsewhere in the sets may contain the same authors who, on that occasion, may be matched. Therefore in the experiments two sets of d-precision and d-recall figures were obtained for the author-matching portion of the system: one for the best overall d-precision/d-recall figures, and another one (with a higher Levenshtein threshold) which resulted in a higher d-precision at the expense of d-recall. Given the need for higher d-precision, it was envisaged that the second of these algorithms would be the one used in a real-world system.

Usefulness

The remaining aim of the experiments described in this chapter is to determine the usefulness or otherwise of the data created by merging the two datasets. Specifically, this questions what changes (if any) would be needed by the client programs in order to take account of the joined data, and asks whether it is sufficient to join just paper IDs and author IDs, leaving other class instances such as citing papers and author affiliations unaffected.

The following section details the results of the experiments and discusses those results.

6.4 Results and discussion

6.4.1 Paper Matching

The matching algorithm for papers, requiring three conditions to be met (identical paper title, paper title containing more than one word and paper title seventeen characters or more in length), yielded 131164 matches out of total dataset sizes of 697959 (ACM) and 574178 (Citeseer). This suggested overlap figures of 18.79% and 22.84% respectively for the ACM and Citeseer datasets.

D-precision was determined by taking a sample of 10000 results (7.6% of the total) and checking these manually. Of that sample, 147 were found to be incorrectly matched papers, yielding a d-precision of 98.53%.

D-recall was harder to determine without manually going through every record in the database. However, the Citeseer team at Penn State University had themselves been working on creating hyperlinks between Citeseer document pages and matched paper pages in the ACM and DBLP portals, and discussions yielded the observation that currently their matching algorithm yielded approximately a 20% overlap of Citeseer's dataset with ACM, but would expect in a fully-working system to yield approximately 30% overlap on the Citeseer side. Taking that 30% as a working estimate, this would yield a d-recall figure of approximately 76%. Although low compared to the d-precision figure, 76% was considered sufficient for use in a prototype system such as this. However, future work should look to incorporate Levenshtein distance algorithms into the programmatic process of paper matching.

The majority of the incorrectly matched results were due one of the key problems with these two particular datasets: Citeseer containing an original conference or journal paper, which was later included as a chapter of a book published by the ACM. While most of these occurrences are picked up by a difference in publication year, a number of records either had no year recorded in the metadata or the ACM publication occurred in the same year as the original paper. In the majority of cases, however, changes (sometimes significant) were made to the papers before their inclusion as book chapters, so therefore they must be considered different instances and should not be mapped. An example of this set of false matches is the paper "Parallel Dynamic Programming" by Zvi Galil and Kunsoo Park, Citeseer reference 43665, later published in the volume "Advances in Parallel Algorithms" in the same year (1992), the book chapter given an ACM reference of 140474. While clearly the same basic paper (and indeed cross-referenced by Citeseer's website), revisions were made and other metadata (such as source and editor of volume) would be different.

The second major group of incorrectly matched results were due to badly parsed data by both Citeseer and ACM. While badly parsed data is quite unusual, these cases are even more unusual as it requires one or both datasets incorrectly parse data such that two differently-titled papers end up with the same title. An example of this is the book chapter with ACM ID 234186 entitled "Neural Networks for Database

Applications" by C. C. Klimasauskas and the paper with Citeseer ID 86256 entitled "Neural Networks for γ/Hadron-Separation with the HEGRA Geiger Towers" by Westerhoff *et al.*. Despite both being published in 1996, they are clearly different papers from different fields. However, parsing errors led to both papers being titled "Neural Networks for" in their respective datasets. Since the year matched, the titles matches, the title contained more than one word and the length of the title was 19 characters, the matching algorithm declared the match a success. While it might appear that attempting to match authors as a confirmation would be a useful additional step to take, it should be noted that a number of papers do not have any author listed (particularly in the Citeseer dataset, where 8.62% of the papers do not have an attributed author) and for papers that appear in the ACM set as part of a book, the 'authors' attributed to the paper are often confused with the overall authors or editors of the volume. Overall, however, this group of incorrectly matched results was sufficiently small (52 out of 10000) to not warrant a great deal of further attention.

The third and final group of incorrectly matched results were papers with the same title that were, genuinely, different papers which happened to share a title. For titles seventeen characters or over in length, this was a small group (9 out of 10000) and its small size serves to back up the choice of seventeen characters as a minimum limit for matching title length. An example of this group would be the papers titled "Binary Decision Diagrams", by Fabio Somenzi with Citeseer ID 329802, and by Minato and Muroga with ACM ID 341258. At 24 letters, this was one of the longest false matches found (the longest being "Parallel Dynamic Programming" at 28 characters). Again, further iterations of the system could be used to improve this figure but the group was so small that the potential improvements were outweighed by other demands on research time. Unlike author matching, false positives in paper matching are, while undesirable, a good deal less problematic.

6.4.2 Author Matching

The matching algorithm for authors was based on string similarity metrics, specifically Levenshtein distance as applied by the SimMetrics package, which yields Levenshtein results normalised between 0 and 1. The key question therefore is to discover the optimum threshold at which a match can be declared. Again, this is measured using d-precision and d-recall metrics. Over the experiments conducted, two particularly interesting thresholds were determined: firstly an optimum threshold yielding highest F-Measure (combination of d-precision and d-recall) and secondly a practical threshold that reflected the importance of having an extremely high d-precision figure, even if this is at the expense of d-recall.

In both cases, d-precision and d-recall results were calculated using a sample of 4000 author matches taken from the 131164 paper matches. The first, optimum, threshold of Levenshtein distance was determined to be 0.51, which yielded d-precision of 98.91% and d-recall of 99.62%, thus yielding the highest possible F-measure of 99.26%. However, a d-precision of 98.91% means that greater than one author match in every hundred is incorrect, which in turn has the knock-on effect of linking together the two sets of documents associated with this author, and producing false results for both authors. A higher threshold was found at a Levenshtein distance of 0.69, which yielded a lower d-recall of 93.02% but a higher d-precision of 99.85%. Although the F-measure was lower at 96.31%, this meant that out of 4000 results, only 6 false author matches were found. Given the additional possibilities of matching the same authors again elsewhere in the system, a d-recall of 93.02% was considered acceptable – certainly more so than having a larger number of false positives. Overall, the difference between the two thresholds was that out of the sample of 4000 results, raising the threshold from 0.5 to 0.69 meant 53 matches were missed but 8 false positives were also excluded.

Raising the threshold still higher would reduce further the number of false positives, although to remove them altogether would require raising the threshold above the highest false match value of 0.75 (Rajiv Gupta and Rajesh Gupta), which would lose a substantial number of correct matches. Lowering the threshold would mean including missed matches such as Carla S Ellis/Carla Schlatter Ellis (0.62) but including more false positive such as Richard P. Graves/Richard F. Rashid (0.65). Even the theoretical low threshold of 0.51 would miss out on a small number of matches such as Dan Ellis/Daniel P. W. Ellis (0.5).

Therefore for a practical system it was determined to set the Levenshtein threshold at 0.69, keeping the number of false positives very low and still producing a reasonably high d-recall figure.

6.4.3 Usefulness of data

Initially, the prototype systems produced matches for just papers. These data were loaded into the knowledge base containing both datasets, and it was found that while paper information and citation counting produced useful data (particularly taking into account the overlap of the two datasets, counting each overlapping citation only once), the non-matching of authors led to false results such as a doubling of the number of authors of each matched paper.

The completed full matching of authors across matched papers produced far more useful results. As well as reducing the number of authors on matched papers down to the actual number, the statistics produced for each author now reflected the works of the authors covered by both datasets. For example, the paper "An Integrated Approach for Improving Cache Behaviour" (ACM paper 1022822, Citeseer paper 568366) links together the ACM author P186192 (Mahmut Kandemir) and Citeseer author 154294 (Mahmut T. Kandemir). This author has 12 papers that are contained in both datasets but overall has 66 papers in the ACM set and 38 papers in the Citeseer set. This linkage allows for joint statistics to be presented for Kandemir's total of 92 distinct papers, including merging of statistics for the 12 overlap papers. A minimum of programmatic changes were required to take account of these overlaps, as described in the following section.

6.4.4 Programmatic Changes

While for the most part very few changes had to be made to the programs to take account of the *owl:sameAs* data, it did become a requirement to add an additional method to the programs concerned as the existing version of 3Store (version 3) did not automatically take into account OWL's inference relations such as *owl:sameAs*. With the additional *owl:sameAs* module in place, the only changes required to be made to the existing code were the encapsulation of the SPARQL queries as calls to the new method rather than directly to the 3Store. The module then returned all suitable results.

It is therefore clear that the system as implemented did produce both an accurate and useful set of crossover data, implemented using the RDBMS version of the data as the source.

6.5 Conclusion

The experiments conducted in this chapter have shown that merging datasets is not only feasible but practical and useful. The data preparation process described in chapter four can therefore be amended such that the matching process is included. The resulting system architecture is fully described in Appendix B and shown diagrammatically in figure B.3.

The usefulness of OWL reasoning is shown by the ease with which the new data could be assimilated into the existing system. While the best source for the data is from within the RDBMS, it is clearly more useful to output the matched data directly into *owl:sameAs* statements and insert into the existing knowledge base. This chapter has shown that an ontology-based data structure is ideally suited to live data, which is often messy and incomplete. Thus the fourth sub-hypothesis is proved.

Chapter 7 Conclusion

The growth of online digital libraries combined with the move towards Open Archiving has raised expectations of what can and should be available in terms of services. While OAI:PMH and the increasing percentage of academic papers available online means that more raw data is available than ever before, the question of how to harness this data and turn it into knowledge is one of the key challenges facing the digital library communities.

At the same time, Semantic Web technologies are developing to the point where large, scalable RDF stores containing many millions of triples have become reality and the development of SPARQL as a standard querying language has meant large-scale standardised semantic web applications are now a possibility. The key challenges in this area are developing such applications to best make use of the extensibility and inference capability offered by ontology-based data sources.

This thesis has considered real-world applications that show the potential for overlap between these two growth areas, and has shown that the Semantic Web offers a new direction for digital library services.

7.1 Conclusions

The central hypothesis of this thesis is that research influence metrics based on semantic bibliometric calculations, performed on ontology-represented data, offer advantages over traditional influence scores. In order to prove this hypothesis, this thesis describes four experiments aimed at proving four sub-hypotheses.

The first sub-hypothesis is that using graph-based approaches to tackle the problem of metadata disambiguation is both useful in terms of semantic bibliometric data preparation and enrichments, and superior to existing approaches.

Chapter three describes a supporting system for disambiguating authors to a high degree of d-precision and d-recall, allowing data to be prepared for use at multiple levels.

Experimentation showed that for citation graph data, author disambiguation can be accurately performed by taking citation data and using it, along with other existing metadata and graph approaches, to match together candidate authors with a degree of success and flexibility greater than others described in the literature. This proves the first sub-hypothesis.

The second sub-hypothesis is that structuring document metadata in an ontology format allows for citation counting and other semiometric calculations at different levels of granularity better than traditional RDBMS/SQL approaches.

Chapter four described a data preparation structure that took this raw metadata, including citation details, and asserted it against a suitable ontology, allowing the systems described in chapter five to function more effectively using SPARQL queries to a 3Store rather than (or alongside) SQL queries to a traditional RDBMS.

Experimentation showed that for most complex, relationship based queries, SPARQL queries were vastly superior to the equivalent SQL queries in terms of time taken to respond. While it was also shown that heavily-indexed RDBMS tables were also capable of producing results in timely fashion, the requirement for easy, quick addition of data to the system meant that the lengthy time taken to perform inserts and updates on a heavily-indexed database rendered such heavily-indexed systems unworkable. The ontology approach combined with SPARQL queries proved the only system capable of producing timely responses to the queries required by the semiometric systems while remaining open to new data being added freely and frequently. In particular, relationship-based queries such as those which add together metric scores to provide results for higher levels of granularity were shown to be more practical in the ontological approach as far less background data preparation had to be performed. This proves the second sub-hypothesis.

The third sub-hypothesis is that metrics more widely ranging than just citation count can be used to determine the influence of a paper, and by inference the author, institution and other levels.

Chapter five described two Semiometric applications built upon semantic web services, allowing ranking according to weighted combinations of citation count, mean citations, authority ratings and, for authors, H-Index and a new modified H-Index which takes into account the authority weighting of each citation as well as the citation count alone.

Experimentation showed that not only were the rankings created by the system found to be at least comparable, and sometimes superior, to those created by citation counting alone, according to expert analysis, but that the variety of rankings that could be created using the ontology-based approach allowed a far more flexible system than a pure citation count system alone. In short, the Semiometric applications described can provide more widely-ranging services than traditional citation count applications, superior to existing systems in the literature, particularly in terms of investigating research at the sub-discipline level. This proves the third sub-hypothesis.

The fourth sub-hypothesis is that amalgamating data from heterogeneous sources (such as harvesting from across the web) is easier to perform when the data is held in an ontological format.

Chapter six describes the changes that need to be made to the data preparation system and to the Semiometric client programs in order for data from different sources to be both merged and queried. In terms of data availability, it was shown that simply adding both sets of data into the same knowledge base provides a single, searchable dataset; in terms of matching overlaps of papers and authors it was shown that simple string matching algorithms with suitable threshold conditions could be used to match papers and authors with a high degree of d-precision and d-recall.

The minimal changes required to the existing system showed the ease with which data from heterogeneous sources could be integrated and used. The most suitable source for the matching data was shown to be the existing RDBMS system, the most suitable output for the matched data was shown to be a simple RDF file containing *owl:sameAs* statements, taking advantage of the inference capabilities offered by OWL. The experiments therefore backed up the approach of joint, synchronized data sources held in RDBMS and ontology formats, and showed that overlap data is best stored and queried using OWL inference statements in ontological format, thus proving the fourth sub-hypothesis.

Taking the evidence shown from the four sub-hypotheses into account, it is clear that the central hypothesis of this thesis has been proved. Representing and manipulating document metadata in an ontological format offers clear advantages over traditional RDBMS approaches.

7.2 Future direction of work

While the system described in this thesis has been used to prove the sub-hypotheses, it is always possible to improve upon existing work. This section details the range of future work that should be undertaken to improve upon the existing system and data structures.

7.2.1 Immediate improvements to process and applications

Most importantly it should be noted how improvements could be made to the existing semiometrics system. The system as it currently stands remains a prototype showing proof-of-concept rather than being a working, real-world system. Several key improvements would be required to make this a fully-operational system.

Firstly, while at present the system produces data based on Computer Science papers from Citeseer and ACM sources, the data is quite incomplete. Citeseer is estimated to cover approximately 40% of the total literature in Computer Science, however even the combination of the Citeseer and ACM datasets do not necessarily give a representative number of citations. For a truly representative, useful system to rival the ISI JIF rankings an acceptable standard of representation would be required.

As the percentage of documents and document metadata available in Open Archive format increases, as well as the increasing coverage of online digital libraries, it is not unreasonable to suggest that a wider-ranging system covering more disciplines with more representative citation results will soon become a realistic possibility.

Secondly, a live, scalable system will require a degree of upkeep: the need to balance background merges and updates as described in chapter six with the availability of live system data in as described in chapters four and five, as well as the ongoing background author disambiguation process from chapter three, would require at the very least two versions of the data store (production and development). Some manual checking of disambiguation and data merging would be required as d-precision in these areas will always be less than 100%. A fully automated system would certainly need to be augmented by at least a small amount of regular human interaction. Additionally, of course, human interaction might be necessary to locate new data for inclusion into the system, although with the increasing development of OAI:PMH this may be able to be automated to a high degree.

Additional, less immediate, changes are also required for a fully operational Semiometrics system, and following sections summarise these.

7.2.2 Ontology Usage

Another potential improvement would be to utilise more of the AKT Reference Ontology and the classes it contains. A study in 2005 of usage of the AKT Reference Ontology showed that only 38.8% of the classes and 50.7% of the properties were actually used in application queries [Alani *et al.* 2005]. The Semiometrics system has already utilised some of the previously unused portions of the ontology through its use of the citation relation 'cites-publication-reference' and related sections. While not all the classes within the AKT Reference Ontology are relevant to the Semiometric system, it is clear that some parts of the ontology, such as those dealing with geographical locations and research interests of groups, could be applied.

7.2.3 Incorporate with OAI:PMH

A major area of extension for the existing system would be to incorporate automated data harvesters, perhaps based on the OAI:PMH protocol. This could be done either through the creation of an OAI:PMH data harvester specifically designed for use within the Semiometrics system, or an existing citation-crawling service such as Citebase [Brody 2003] could be incorporated to use both its targeting capabilities and its data in the Semiometrics context.

7.2.4 More measures, more levels

As has been mentioned throughout this thesis, the system as it currently stands is a proof-of-concept prototype rather than a fully operational system. As such, it provides relatively few measures across only three levels of granularity (paper, author, sub-discipline). Future iterations of the system should include, as stated in section 5.5.3, other measures such as the G-Index to account for extremely highly cited papers, along with a weighting on paper categorisation to give more emphasis to a paper's primary categorisation. Alongside this, future disambiguation work should serve to allow extra levels of granularity, including research groups, institutions, countries and journals/conferences, allowing relative ranking scores for each of these areas of interest.

7.3 The future and Semantic Digital Libraries

It is clear that there is a demand and an expectation for large-scale digital libraries and related services. The success of repositories and tools such as arXiv, Citeseer and more recently Google Scholar, combined with OAI and the drive towards open access to all academic output, shows clear and growing momentum towards fully online, accessible digital libraries of scientific papers. At the same time, the growth in effectiveness of scalable semantic web systems allows an effective ontological approach to be taken on digital library metadata containing tens of millions of triples. While the future of the Semantic Web itself is as yet unclear, it is certain that expectations of services in the Digital Libraries domain are rising, and the ontological approach, based on Semantic Web technologies as described in this thesis, provides a sensible, scalable solution to the need for more intelligent data and knowledge handling.

Appendix A

Summary of Reviewed Systems

This appendix summarises the various Digital Library and Document Management systems reviewed in this thesis. The systems are taken from across application domains to show different expectations and requirements as well as differences in pure functionality. Although quantitative analyses of these systems is often hard, especially when underlying techniques and architectures aren't known in detail, quantitative results are described where possible, thus allowing comparisons to be made. It is also important to note that this list is by no means exhaustive: other DLs exist of the various types such as the IEEE Digital Library in the publisher field, DSpace in the self-archiving repository field and Documentum in the commercial field. There are also numerous applications at the fringe of the DL definition, for example open-source Content Management Systems such as [Drupal] and [Mambo].

Each system was evaluated considering:

1. Overall metadata model
2. Document submission
3. Versioning
4. Metadata capture
5. Metadata storage
6. Metadata correction
7. Disambiguation of authors and other metadata fields
8. Search techniques/interface
9. Search results (incl. document availability)
10. System capacity
11. Degree of automation
12. Distributed repositories
13. Citation analysis
14. Security model (at user level, incl. subscriptions)

Google Scholar

Built on top of the Google web-crawling search engine, Google Scholar was released in November 2004 and allows users to perform familiar Google-style searches (eg single-line full-text index searching, metadata filtering eg author:Shadbolt) over documents determined by Google to be academic papers. It allows more domain-specific metadata and searches than the standard Google service and returns results (including citations in and out) from a variety of sources, including papers available only by subscription.

1. Citations, title, author, date, source; also full-text indexing.
2. Crawling via Google.
3. No.
4. Automatic via format-reading techniques.
5. Google repository.
6. Unknown.
7. Name-based.
8. Single-line search incl. allowance of metadata within.
9. Some documents unavailable, dependent upon user subscriptions. Results warn of unavailable documents.
10. Unknown.
11. Fully automated.
12. Yes.
13. Yes, although uncertain about de-dupe or virtual documents.
14. Returns results of unavailable documents due to subscriptions. Security of subscriptions left up to source website.

Citeseer

Created at NEC labs and fully described in [Lawrence *et al.* 1999], Citeseer introduced the concept of Autonomous Citation Indexing (ACI), where papers (including those in PDF and PS formats) were fully indexed. Metadata and bibliographic information were extracted, allowing citation links between papers to be rendered as hyperlinks on the web-based front-end. A bespoke Perl-hash database with a directed metadata model provided the back-end as, at the time of creation, standard SQL databases were not capable of providing the speed of response required by the system. Citeseer is currently owned and maintained by Penn State University and has several mirrors around the world, including one at Southampton University, installed as part of the work described in this thesis.

1. Dublin Core, citations, context (incorporating full-text indexing).
2. Crawling or manual targeting following suggestion to admin.
3. No.
4. Automatic via format-reading techniques.
5. Bespoke Perl-hash DB, also periodically exported to XML in OAI-compliant format.
6. Email feedback to global admin.
7. No.
8. Single-line search on full-text index, title, author.
9. All returned documents available from local cache.
10. >700 000.
11. Targeted harvesting, metadata correction; everything else automated.
12. Experimental.
13. Fully implemented incl. de-dupe and 'virtual documents' as yet unharvested.
14. No.

Rexa

Released in April 2006, Rexa is the product of research at the Information Extraction and Synthesis Laboratory at the University of Massachusetts. Based on a similar model to that of Citeseer (automated scraping, indexing and citation analysis and a web-search front end), it is particularly concerned with the creation of widely-ranging data models based on the scraped papers. In particular, disambiguation of concepts such as document author, institution and document topic give Rexa a rich source of data. While the approaches used by the Rexa group do not appear to include semantic representations, there are clear overlaps with some of the questions raised in this thesis (particularly the questions of data disambiguation raised in chapters three and six) and as Rexa develops as a valid data source, the expression of their data in semantic format looks as if it would prove a worthwhile exercise.

1. Dublin Core, citations in/out, user tagging, extracted topics, grants. More promised in the future, incl. research communities, institutions, conferences and journals.
2. Targeted crawler-based database population; metadata and link submission page available but not yet linked to system.

3. No.
4. Automated, populating about fourteen different bibliographic fields via second-order linear-chain conditional random field information extraction methods.
5. Standard SQL DB, full-text indexing using Lucene open-source indexer.
6. None except via future iterations of the system.
7. Automated cross-referencing on authors and other fields using partitioning methods, parameterised edge weights, and parallel processing on over 100 compute servers. Also topic discovery using a phrase-aware variant of Latent Dirichlet Allocation.
8. Single-line search on any metadata area or full-text index.
9. Documents sometimes available depending on available links and cached version.
10. Currently >300000 documents in cache, 7 million virtual documents.
11. Fully automated with possible exception of targeting spider.
12. No.
13. Fully implemented.
14. User account login, no other security. All features available to all users.

DBLP

Originally created as a server store for papers on DataBase systems and Logic Programming, DBLP has now expanded across the Computer Science field and its acronym is now said to mean 'Digital Bibliography & Library Project'. Largely manually maintained by creator Michael Lay and colleagues from the University of Trier, DBLP is widely regarded as a reliable, wide-ranging source of information on papers and authors in computer science. DBLP is hosted on a number of mirror servers, including the ACM, and its >850 000 document records are also available for XML download. Future iterations of the Semiometrics systems look to include DBLP as an additional data source.

1. Dublin Core plus a few extra fields. A subset of papers include citation links.
2. Automated discovery. Email submission of journals/conferences is offered but not guaranteed – the more votes, the more likely it will be accepted. Emailed single documents highly unlikely to be accepted.
3. No.

4. Document and journal/proceedings data provided by submission or discovered from source. Individual papers not parsed.
5. Data held in flat-file format (XML and similar). Hand-created tools (C, Perl, Java) create and maintain these files.
6. Manually, via email requests. Not guaranteed due to limited resources.
7. Partly manual, partly via specifically tailored program 'mkauthors'.
8. Single-line search on title and author, advanced search on other metadata fields, including multiple authors.
9. Metadata returned, including link to 'author' pages if they exist. For some papers, 'Electronic Edition' ('EE') links take user to a page with more information, or link to the document held on an external server. No documents are available in a public cache.
10. >850 000.
11. Maintenance of much of the data performed by hand; apart from that, creation process is largely automated from supplied contents data for journals, proceedings and books.
12. Synchronised mirror sites, although all contain identical rather than distributed data.
13. A subset of the documents have bibliography analysed to allow links to other DBLP documents.
14. None required: metadata search facility is publicly available; no papers are provided.

ACM Digital Library

Slightly different to those listed above, the ACM Digital Library allows search and browse access to all ACM-published items. Specifically it includes conference proceedings, articles from periodicals, divisions (usually chapters) of books and whole books. A subscription-based model is applied where members and subscribers are allowed download access to appropriate content; non-members may use basic search and browse facilities and purchase articles for a fee. All items in the ACM DL are classified, at time of submission, by the authors against the ACM Computing Classification System, last reviewed in 1998. Each item may be classified against more than one category.

1. Dublin Core + categorisation + citations.
2. Automated for all documents published by ACM.

3. No; no need (final published version only). When development of paper published in different ACM location, has own identity.
4. Automated due to standard format.
5. Uncertain; data exported in XML.
6. Not required; all metadata based on user-submitted information.
7. Yes.
8. Single-line search, more metadata-specific available.
9. Subscriber model. Available documents returned from local cache.
10. >700 000.
11. Based on data entry at time of publication.
12. Unknown.
13. Fully implemented internally.
14. Requires subscription to ACM.

EPrints

A repository system tailored for institutional usage, EPrints shares many features with the similarly-focused DSpace. Building on the move towards Open Archiving, EPrints not only looks to provide a technical solution but also actively promotes users publishing their research online through both the web portal interface and the ability to export metadata in OAI-compliant format. Currently on version 3, the team is based at the University of Southampton, although there were no direct links between the work described in this thesis and the EPrints project. Future integration of OAI-compliant metadata into the Semiometrics system would allow for direct, simple integration with EPrints and future iterations should take this into account.

1. Mix of user-created and system standard per installation. Standard fields created through collaboration with institutional library specialists. Full-text not supported internally but options such as 'htdig' software and Google site search are suggested.
2. Manual upload via web interface. Optionally users can choose to upload a link to another location rather than the document itself.
3. Yes.
4. Entered by user at time of upload.
5. MySQL backend database.
6. Manual correction through account-based web interface.
7. Author accounts allow for user-performed manual disambiguation.

8. Single-line search on metadata (full-text can be integrated; see point 1 above). Advanced metadata search and browse capabilities available.
9. Metadata returned for paper, including links to author page and related documents. Full document available from repository or via uploaded link, security model allowing (see point 14 below).
10. No documented figures. Would appear to be limited only by back-end database capabilities.
11. Installation required per repository, along with regular system maintenance. Author-maintenance of repository contents. Metadata exporting fully automated.
12. A single instance of EPrints can run multiple repositories; each repository is independent rather than distributed.
13. Not directly. Related Citebase work [Brody 2003] covers this area.
14. User registration for document submission. Optionally, system is browseable only by registered users. Subscription model supported.

Stellent

Formerly known as IntraDoc, the Stellent 'Enterprise Content Management' system is at its core a document management system. Created in 1996 and built entirely using Java, the system is geared towards internal document storage for business customers, although web publication is also included as a standard model. Having over 4700 customers worldwide, Stellent were acquired by Oracle in November 2006 for $440 million and future iterations of Stellent are likely to be closely tied to Oracle's database model, including their 10g RDF triplestore module.

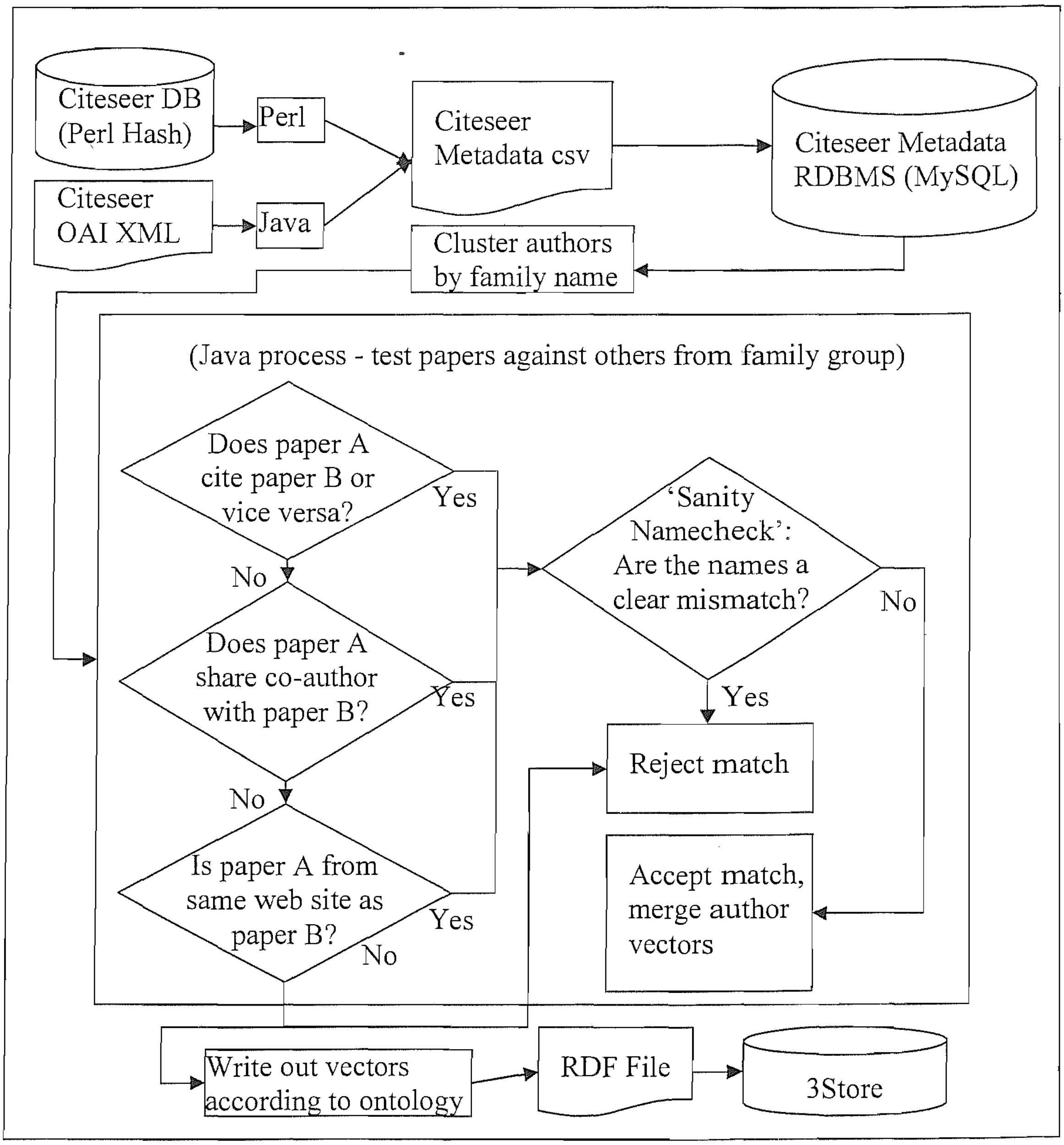
1. Mostly user-determined; basic DC default (dDocAuthor, dDocTitle, dInDate). Full-text performed by 3rd party (Verity) search engine.
2. User-driven submission (via workflow)
3. Yes, incl. rollback.
4. User-entered per document/per batch.
5. SQL-based RDBMS
6. Any allowed system user
7. System users only
8. Single-line search on full-text index; also field-by-field metadata (can be customised)
9. All documents available within constraints of security model.

10. >1 to 2 million with default setup. Can be much greater (>10 million, perhaps >50 million) dependent on distributed instances, metadata structure, underlying OS and 3rd party search/index engine (default engine, Verity, shows degradation as document count approaches 1 million).
11. User-event driven (submission, approval, metadata annotation); some parts automated (indexing, actual publishing, expiry/renewal process).
12. Yes.
13. Not by default (could be coded using dDocID and manually-entered; no ACI or other text-analysis tools included by default; again, ACI potentially could be created as component).
14. Two-dimensional: user security clearance and group membership.

Appendix B

Technical Overview of Semiometrics System

This appendix gives a technical summary of the components that make up the Semiometrics system, as used by the experiments described in this thesis. Firstly, chapter three describes AKTiveAuthor, a system for taking raw metadata from and performing analysis techniques to determine if the authors are the same person. The following diagram describes the implementation of the system:



Citeseer DB (Perl Hash)
Perl
Citeseer OAI XML
Java
Citeseer Metadata csv
Citeseer Metadata RDBMS (MySQL)
Cluster authors by family name
(Java process - test papers against others from family group)
Does paper A cite paper B or vice versa?
Yes
No
Does paper A share co-author with paper B?
Yes
No
Is paper A from same web site as paper B?
Yes
No
'Sanity Namecheck': Are the names a clear mismatch?
No
Yes
Reject match
Accept match, merge author vectors
Write out vectors according to ontology
RDF File
3Store

Figure B.1 : AKTiveAuthor Implementation Details

Fig B.1 shows the processes described in chapter three. From an implementation point of view, most of the work is done using Java, with the exception of the initial data extraction. As is shown, this initial extraction of data is done either directly from the Citeseer database via Perl or, using Java JAXP XML parsing techniques, from the OAI XML files published by the Citeseer team at Penn State periodically. The output csv file is then imported into a relational database (in our case SQL) according to the structure shown in chapter four, fig 4.1. Searches are performed on this to cluster sets of papers by family name (for example, all papers authored by someone named 'Smith' or 'Anderson') and within these clusters, each paper is tested against the others according to the criteria set out above. Initially, each author is a Vector of size one, but the idea is that as matches are found, we merge the Vectors to reflect the papers we have found with matching authors. Candidate matches are verified using the 'sanity namecheck' to see whether the author name is obviously not correct (eg attempting to match 'A. J. Smith' to 'Brian H. Smith') and if a candidate match passes this test, the contents of the Vector of the second author are added to the Vector of the first, and the second Vector is then discarded. Once all the papers for a name-cluster have been processed in this way, the final set of Vectors can be written out to RDF and asserted in 3Store.

In the specific implementation described in chapter three, the RDF files are actually created at the end of the whole process rather than after each name-cluster is processed; additionally the joined-up author data is fed back into the RDBMS, into appropriate tables. Neither of these factors affects the performance of the system. The main matching program, written fully in Java, was run at various times on both a desktop Windows machine and a high-end 64-bit Linux server, using identical code, and performance in both cases was acceptable. The 3Store and RDBMS were housed on the server in both cases, queries being performed directly or via JDBC/HTTP as appropriate.

Secondly, chapters four and five describe the Semiometrics system as used in the respective experiments to test system feasibility and influence measures respectively. The following diagram shows the implementation details of the processes described in those chapters:

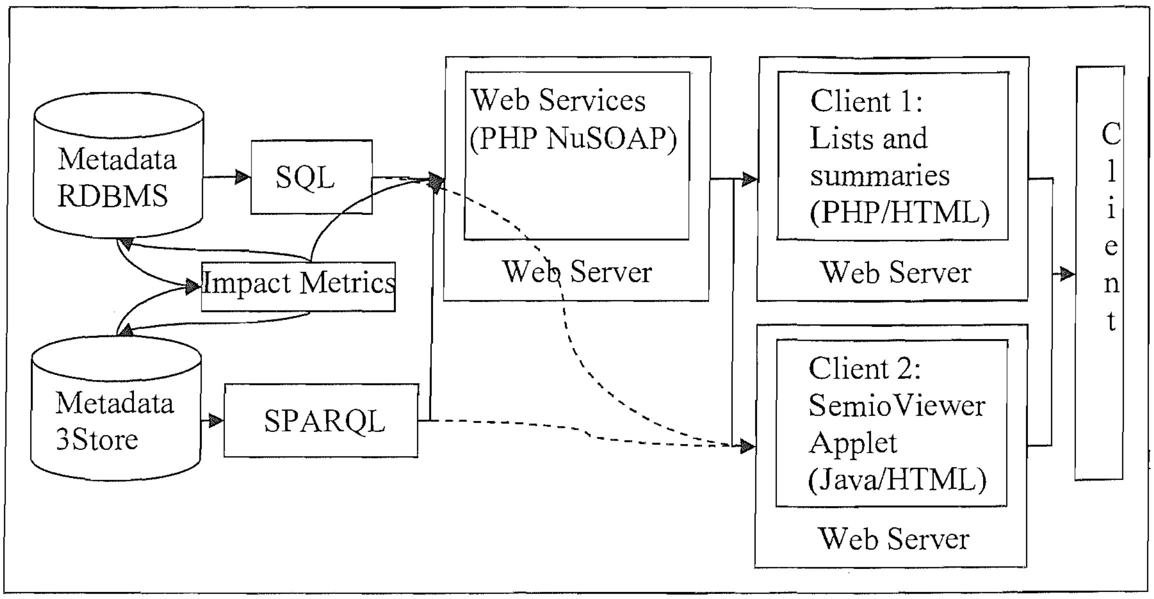

Metadata RDBMS
SQL
Impact Metrics
Metadata 3Store
SPARQL
Web Services (PHP NuSOAP)
Web Server
Client 1: Lists and summaries (PHP/HTML)
Web Server
Client 2: SemioViewer Applet (Java/HTML)
Web Server
Client

Figure B.2 : Semiometrics system implementation details

The Semiometrics system, as described in chapter four, necessarily takes its data from two sources, the traditional RDBMS (MySQL) and the RDF triplestore. In most cases, a set of web services performs the SQL and SPARQL queries as appropriate, either directly (SQL) or over HTTP (SPARQL). These web services are written in PHP and are built using the open-source [NuSOAP] toolkit which supports SOAP 1.1 and WSDL 1.1. Network-based calculations (influence metrics), such as those determining hubs-and-authorities, betweenness centrality and PageRank scores, are performed either at run-time or, more usually, as a background batch-process and re-stored (effectively cached) in the database and/or triplestore. These data are converted to simple Pajek net format, and calculations are performed using a Java application invoking the JUNG API. The services shown are those required for the semiometrics system to function, and for each service there is both a conventional (SQL) version and a semantic (SPARQL) version, the latter having the service suffix '3s3' to represent its compatibility with the SPARQL-compliant 3Store version 3 release.

As summarised in chapters four and five, the first client system is also PHP-based and performs inference calculations on paper and author data to determine ranking lists and influence scores for these levels. The web server on which the client systems appear is typically the same Apache installation as the one running the web services, although during the experiments a Windows desktop version of Apache was also tested successfully for the clients with identical code. The second client performs largely the same calculations as the first but allows visualisation of networks (typically co-authorship networks for authors and citation networks for papers) rather than creation of ordered lists. The sub-discipline calculations in chapter five were performed using a combination of both clients. The dotted lines indicate part of the experimental process, where the SemioViewer applet was directly connected to the data sources via SQL and SPARQL (using tailored java client programs to mediate the queries and responses). These were designed to test whether direct connections showed any differing performance in response times for complex querying. No notable difference was found, although the querying was understandably simpler as it required only direct SQL/SPARQL rather than a full SOAP approach. Both approaches remain coded into the second (SemioViewer) client. When the *owl:sameAs* data from chapter six was included in the system (see below), it became necessary to use the direct connection approach for RDF queries as a reasoning tool had to be created to perform the inference calculations and this was most easily created by adding a method to the tailored java client program for RDF querying. Such functionality should be added to the web services client in future work.

Finally, chapter six describes amendments to the system data flow process required to incorporate heterogeneous data sources into the Semiometrics system. The following diagram summarises the new system structure required to allow semantic integration of data utilising basic OWL Lite capabilities, specifically the *owl:sameAs* function.

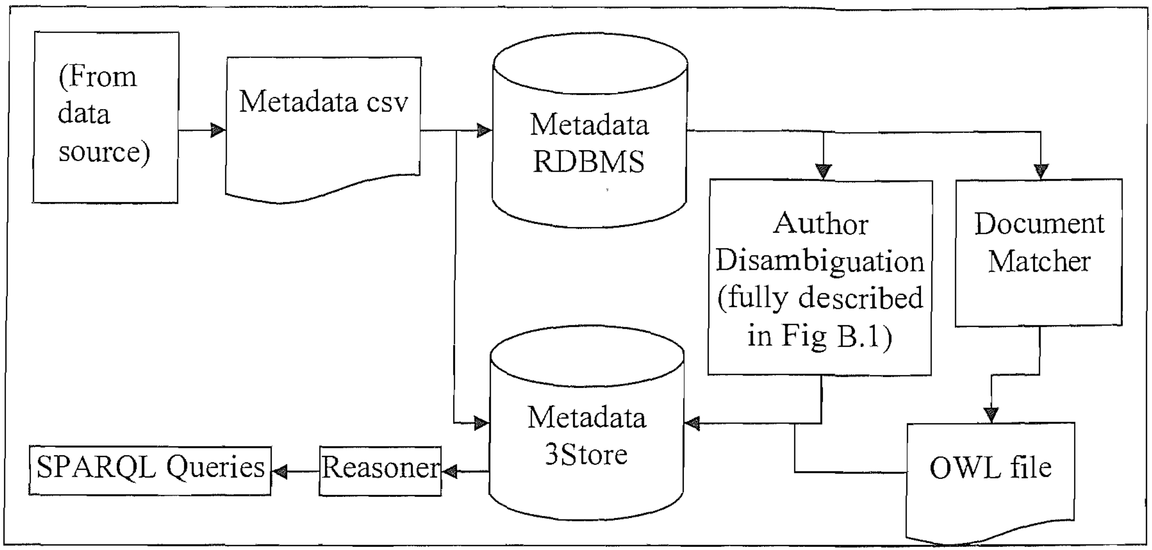


Figure B.3 : System amendments to allow for semantic data integration

Note: in the context of this work as a disambiguation tool, it would be reasonable to extend this to include use of *owl:differentFrom* to express explicit disjoints in instance data; however this is beyond the scope of the requirements here and as such was not implemented.

As covered in chapter six, for performance reasons it is necessary to run the document matching process from the RDBMS version of the metadata store rather than the RDF version as there are a large number of identity-centric queries required by the process, which (as proven in chapter four) are handled better by the traditional RDBMS approach. As also stated in chapter six, the document matching process is performed using Java rather than Perl due to the use of the SimMetrics package, despite the added complexity when performing regular expression matching. The 'reasoner' block that now exists between SPARQL queries and the 3Store is added due to the requirement of having to perform additional calculations to search on data expressed using *owl:sameAs* – 3Store does not perform this inference by default. In the experiments described in chapter six, a small purpose-built reasoner method was added to the RDF client java program which would look for all *owl:sameAs* relationships and search on those URIs as well as those specifically stated in the query. As stated above, this meant that the SemioViewer application would be restricted to direct SPARQL queries rather than going through web services, although as SPARQL queries are performed over HTTP this is not a limiting factor from a performance point of view.

However, it is worth noting that the created reasoning method is extremely limited: it is tailored specifically to search for *owl:sameAs* relations for certain metadata types within the AKT Reference Ontology as extended for use within the Semiometrics system. In particular, this means that all *owl:sameAs* relations are fully investigated, which would not be practical if there were a large number of such relations discovered. If a more generic solution were required, future work should take into account solutions such as [Pellet], a generic open-source OWL-DL reasoner written in Java by the University of Maryland.

In addition to the points above, the work described in this thesis required an ontology against which document data could be classified. For the most part, the existing AKT Reference Ontology was used unaltered as it was designed to accurately describe the academic research domain, including publications, journals and researchers. However, a small number of extensions were required to this ontology in order to meet some of the requirements of the semiometrics applications. For example, while the original ontology allowed for researchers to have 'research area of interest' as a specific property (with a range of the top two tiers of the ACM classification), academic papers had no such space to describe their subject areas. Additional extensions allowed papers to have an influence score and an authority score, reflecting the RDBMS structure shown in chapter four, fig. 4.1. The specific extensions to the AKT reference ontology are as follows:

```
   <owl:DatatypeProperty rdf:about="#has-impact-factor">
     <rdfs:label>has impact factor</rdfs:label>
     <rdfs:comment>Indicates that a publication has the given
impact factor.</rdfs:comment>
     <rdfs:domain rdf:resource="#Journal"/>
     <rdfs:range rdf:resource="&xsd;decimal"/>
     <rdfs:isDefinedBy rdf:resource="&extbase;"/>
    </owl:DatatypeProperty>

    <owl:DatatypeProperty rdf:about="#has-authority">
     <rdfs:label>has authority</rdfs:label>
     <rdfs:comment>Indicates that a publication has the given
authority.</rdfs:comment>
     <rdfs:domain rdf:resource="#Publication-Reference"/>
     <rdfs:range rdf:resource="&xsd;decimal"/>
     <rdfs:isDefinedBy rdf:resource="&extbase;"/>
    </owl:DatatypeProperty>
```

```
        <owl:ObjectProperty rdf:ID="has-acm-research-interest">
          <rdfs:label>has acm research interest</rdfs:label>
          <rdfs:comment>Indicates   the   publication   deals   with   a
particular area of ACM research interest</rdfs:comment>
          <rdfs:domain rdf:resource="#Publication-Reference"/>
          <rdfs:range rdf:resource="#Research-Area"/>
          <rdfs:isDefinedBy rdf:resource="&extbase;"/>
        </owl:ObjectProperty>
```

No such extensions to the author/person level objects were required as the purpose of the project was to be able to infer different granularity levels (such as document author) from basic paper information. Such extensions could be added if future iterations of the system required it (for example, if some kind of caching system were to be introduced).

References

[ACM 1998] The ACM Computing Clasification System (1998 Version). http://www.acm.org/class/1998/

[ACM DL] The ACM Digital Library. http://portal.acm.org/

[AKT Reference Ontology] *The AKT Reference Ontology* (2002). http://www.aktors.org/publications/ontology/

[Alani *et al.* 2002] Alani, H., O'Hara, K., Shadbolt, N. (2002) *ONTOCOPI: Methods and Tools for Identifying Communities of Practice,* In proceedings of the IFIP 17th World Computer Congress (WCC), Montreal, Canada, pp. 225-236, 25-30 August 2002.

[Alani *et al.* 2002a] Alani, H., Dasmahapatra, S., Gibbins, N., Glaser, H., Harris, S., Kalfoglou, Y., O'Hara, K. and Shadbolt, N. (2002) *Managing Reference: Ensuring Referential Integrity of Ontologies for the Semantic Web*, In proceedings of 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02), Sigenza, Spain, pp. 317-334, 1-4 October 2002.

[Alani *et al.* 2005] Alani, H., Harris, S. and O'Neil, B. (2005) *Ontology Winnowing: A Case Study on the AKT Reference Ontology*, In proceedings of IEEE International Conference on Intelligent Agents, Web Technology and Internet Commerce, Vienna, Austria, 28-30 November 2005.

[arXiv] Archive and distribution server for research papers. http://arxiv.org/

[Berners-Lee *et al.* 1998]

Berners-Lee, T. (1998) *Semantic Web Roadmap*, W3C, September 1998.
http://www.w3.org/DesignIssues/Semantic.html

[Bjelobrk & Zukerman 2005]

Bjelobrk, I., Zukerman, M. (2005) *Electrical Engineering Research Evaluation*, University of Melbourne Report, 31 October 2005.
http://www.ee.unimelb.edu.au/staff/mzu/igor_report.pdf.

[Bollen *et al.* 2005] Bollen, J., Van de Sompel, H., Smoth, J. A., Luce, R. (2005) *Toward alternative metrics of journal impact: A comparison of download and citation data*, Information Processing and Management, 41:6, pp. 1419-1440, December 2005.

[Boyack 2004] Boyack, K.W. (2004) *Mapping knowledge domains: Characterizing PNAS*, Proceedings of the National Academy of Sciences, 101 (Suppl. 1), pp. 5192-5199, April 2004.

[Braintrack] Braintrack University Index. http://www.braintrack.com/

[Brandes 2001] Brandes, U. (2001) *A faster algorithm for betweenness centrality*, Journal of Mathematical Sociology, 25:2, pp. 163-177, 2001.

[Brody 2003] Brody, T. (2003) *Citebase Search: Autonomous Citation Database for e-Print Archives*, In proceedings of the Third International Technical Workshop and Conference of the project SINN, Oldenburg, Germany, 17-19 September 2003.

[Case & Higgins 2000]

Case, D.O., Higgins, G.M. (2000) *How can we investigate citing behaviour? A study of reasons for citing literature in communication*, Journal of the American Society for Information Science, 51:7, pp. 635-645, April 2000.

[Chalmers 2005] Chalmers, M. (2005) *Five papers that shook the world,* Physics World, 18:1, January 2005.

[Chapman 2004] Chapman, S. (2004) *SimMetrics*.
http://sourceforge.net/projects/simmetrics/.

[Chen 2004] Chen, C. (2004) *Searching for intellectual turning points: Progressive knowledge domain visualization*, Proceedings of the

National Academy of Sciences, 101(Suppl. 1), pp. 5303-5310, April 2004.

[Chen 2006] Chen, C. (2006) *CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature*, Journal of the American Society for Information Science and Technology, 57:3, pp. 359-377, February 2006.

[Citeseer/ACM] http://citeseer.ist.psu.edu/announcements.html

[Citeseer Author Statistics]
http://citeseer.ist.psu.edu/allcited.html

[ClassAKT] Ball, S. (2005) *ClassAKT: A text classification web service for classifying documents according to the ACM Computing Classification System*, AKT Consortium. http://www.aktors.org/technologies/classakt/

[Councill & Giles 2004]
Councill, I. G., Giles, C. L. (2004) *Who gets acknowledged: Measuring scientific contributions through automatic acknowledgement indexing*, Proceedings of the National Academy of Sciences, 101:51, pp. 17599-17604, 21 December 2004.

[Councill *et al.* 2005] Councill, I. G., Giles, C. L., Han, H., Manavoglu, E. (2005) *Automatic acknowledgement indexing: expanding the semantics of contribution in the Citeseer digital library*, In proceedings of the 3rd international conference on Knowledge capture table of contents, Banff, Alberta, Canada, pp. 19-26, 2-5 October 2005.

[Cronin 1984] Cronin, B., (1984) *The citation process*, London: Taylor Graham.

[CrossRef] CrossRef: The Citation Linking Backbone. http://www.crossref.org/

[Darwin 1859] Darwin, C. R. (1859) *On the Origin of Species*, London: John Murray.

[Dong *et al.* 2005] Dong, P., Luh, M. & Mondry, A. (2005) *The Impact Factor Revisited*, Biomedical Digital Libraries, 2:7, 5 December 2005.

[DSpace] The DSpace digital repository system. http://www.dspace.org/

[Drupal] Open-Source Content Management Platform. http://drupal.org/

[Dublin Core] Weibel, S., (1998) *The Dublin Core: A simple content description format for electronic resources*, NFAIS Newsletter: 40:7, pp. 117-119.

[Egghe 2006] Egghe, L. (2006) *Theory and practice of the g-index*, Scientometrics, 69:1, pp. 131-152, April 2006.

[Einstein 1905] Einstein, A. (1905) *On the Electrodynamics of Moving Bodies,* Annalen der Physik, 17, pp. 891-921, 26 September 1905.

[Einstein 1905a] Einstein, A. (1905) *On the Movement of Small Particles Suspended in Stationary Liquids Required by the Molecular-Kinetic Theory of Heat*, Annalen der Physik, 17, pp. 549-560, 18 July 1905.

[Einstein 1949] Einstein, A. (1949) *Why Socialism?* Monthly Review, 1:1, May 1949.

[EPrints] EPrints for digital repositories. http://www.eprints.org/

[Gabehart 2005] Gabehart, M.E. (2005) *An analysis of citations to retracted articles in the scientific literature,* Master's Paper for the M.S. in L.S degree, University of North Carolina.

[Garfield 1955] Garfield, E. (1955) *Citation Indexes for Science: A New Dimension in Documentation through Association of Ideas*, Science, 122:3159, pp. 108-111, July 15 1955.

[Garfield 1972] Garfield, E. (1972) *Citation Analysis as a Tool in Journal Evaluation*, Science, 178:4060, pp. 471-479, 3 November 1972.

[Garfield 1994] Garfield, E. (1994) *The impact factor*, Current Contents, 25, pp. 3-7, 20 June 1994.

[Garfield 1996] Garfield, E. (1996) *Fortnightly Review: How can impact factors be improved?* British Medical Journal, 313, pp. 411-413, 17 August 1996.

[Google Scholar] Scholarly literature search tool. http://scholar.google.com/

[Gruber 1993] Gruber, T. R., (1993) *A translation approach to portable ontologies*, Knowledge Acquisition, 5:2, pp. 199-220.

[Han *et al.* 2004] Han, H., Giles, C. L., Zha, H., Li, C., Tsioutsiouliklis, K., (2004) *Two supervised learning approaches for name disambiguation in author citations*, In proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital libraries, Tuscon, AZ, USA, pp. 296-305, June 2004.

[Han *et al.* 2005] Han, H., Giles, C. L., Zha, H. (2005) *Name disambiguation in author citations using a K-way spectral clustering method*, In proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries, Denver, CO, USA, pp. 334-343, June 2005.

[Harnad *et al.* 2003] Harnad, S., Carr, L., Brody, T., Oppenheim, C. (2003) *Mandated online RAE CVs Linked to University Eprint Archives*, Ariadne, 35, 30 April 2003.

[Harris & Gibbins 2003]

Harris, S. W., Gibbins, N. M. (2003) *3store: Efficient Bulk RDF Storage*, In Proceedings of 1st International Workshop on Practical and Scalable Semantic Systems (PSSS'03), Sanibel Island, Florida, pp. 1-15, 20 October 2003.

[HEIR] Higher Education Institution Registry. http://www.siu.no/heir/

[HESA] Higher Education Statistics Agency. http://www.hesa.ac.uk/

[Hirsh 2005] Hirsch, J. E. (2005) *An index to quantify an individual's scientific research output*, Proceedings of the National Academy of Sciences, 102:46, pp. 16569-16572, 15 November 2005.

[HP 2003] Hewlett-Packard Labs, (2003) *The Jena Semantic Web Toolkit*, http://www.hpl.hp.com/semweb/jena.htm

[JUNG] Java Universal Network/Graph Framework. http://jung.sourceforge.net/

[Kleinberg 1998] Kleinberg, J. M. (1998) *Authoritative sources in a hyperlinked environment*, In proceedings of ACM-SIAM Symposium on Discrete Algorithms, pp. 668-677, January 1998.

[Kleinberg 1999] Kleinberg, J. M. (1999) *Hubs, authorities, and communities*, ACM Computing Surveys 31:4, Article 5, December 1999.

[Kleinberg 2002] Kleinberg, J. (2002) *Bursty and Hierarchical Structure in Streams*, In proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, pp. 91-101, 23-26 July 2002.

[Kuhn 1970] Kuhn, T.S. (1970) *The structure of scientific revolutions*, 2 ed., Chicago: University of Chicago Press.

[Lawrence *et al.* 1999] Lawrence, S., Bollacker, K., Giles, C.L. (1999) *Digital Libraries and Autonomous Citation Indexing,* IEEE Computer, 32:6, pp. 67-71, June 1999.

[Lawrence *et al.* 1999a]

Lawrence, S., Bollacker, K., Giles, C.L. (1999) *Indexing and Retrieval of Scientific Literature*, In proceedings of Eighth International Conference on Information and Knowledge Management (CIKM 99), Kansas City, MO, USA, pp. 139-146, 2-6 November 1999.

[Lawrence 2001] Lawrence, S. (2001) *Online or invisible,* Nature, 411:6837, p. 521, 31 May 2001.

[Levenshtein 1965] Levenshtein, V. I. (1965) *Binary codes capable of correcting deletions, insertions and reversals*, Doklady Akademii Nauk SSSR, 163:4, pp. 845-848, August 1965.

[Malin 2005] Malin, B. (2005) *Unsupervised name disambiguation via social network similarity,* In proceedings of Third Workshop on Link Analysis, Counterterrorism and Security, Newport Beach, CA, USA, pp. 93-102, April 2005

[Mambo] Mambo Content Management System. http://www.mamboserver.com/

[Mane & Börner 2004]

Mane, K. K., Börner, K. (2004) *Mapping topics and topic bursts in PNAS*, Proceedings of the National Academy of Sciences, 101 (Suppl. 1), pp. 5287-5290, April 2004.

[McBride *et al.* 2004] McBride, B., Carroll, J., Klyne, G., (Eds.) (2004) *Resource Description Framework (RDF): W3C Technical Recommendation*, W3C. http://www.w3.org/RDF/

[McCallum 2006] McCallum, A., Mann, G. S., Mimno, D. (2006) *Bibliometric impact measures leveraging topic analysis*, In proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, Chapel Hill, NC, USA, pp. 65-74, June 11-15 2006.

[McGuiness & Van Harmelen 2004]

McGuinness, D. L., van Harmelen, F., (Eds) (2004) *OWL Web Ontology Language: W3C Technical Recommendation*, W3C. http://www.w3.org/TR/owl-features

[McRae-Spencer & Shadbolt 2006]

McRae-Spencer, D., Shadbolt, N. (2006) *Also By The Same Author: AKTiveAuthor, a Citation Graph Approach to Name*

Disambiguation, In proceedings of 6th ACM/IEEE-CS Joint Conference on Digital Libraries, Chapel Hill, North Carolina, USA, pp. 53-55, 11-15 June 2006.

[McRae-Spencer & Shadbolt 2006a]

McRae-Spencer, D., Shadbolt, N. (2006) *Semiometrics: Applying Ontologies across Large-Scale Digital Libraries*, In proceedings of Second International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006), Athens, Georgia, USA, 5 November 2006.

[McVeigh 2004] McVeigh, M.E. (2004) *Open Access Journals in the ISI Citation Databases: Analysis of Impact Factors and Citation Patterns*, Thomson Corporation.

[Millard 2004] Millard, I. (2004) *Ambient Intelligence: The contextually aware environment*, Transfer Minithesis, ECS School, University of Southampton.

[Moed 2005] Moed, H. F. (2005), *Citation Analysis in Research Evaluation* Netherlands: Springer, Dordrecht.

[MSN Academic Live]

Academic publication seach tool.
http://search.live.com/results.aspx?scope=academic

[Myers 2003] Myers, P. (2003) *The Apotheosis of Albert Einstein*, http://users.cyberone.com.au/myers/einstein.html.

[Newman 2004] Newman, M.E.J. (2004) *Coauthorship networks and patterns of scientific collaboration*, Proceedings of the National Academy of Sciences, 101 (Suppl. 1), pp. 5200-5205, April 2004.

[NuSOAP] NuSOAP: SOAP Toolkit for PHP.
http://sourceforge.net/projects/nusoap/

[OAI] Lagoze, C., Van de Sompel, H. (2001) *The Open Archives Initiative: Building a low-barrier interoperability framework*, In proceedings of the First ACM/IEEE Joint Conference on Digital Libraries, Roanoke VA, USA, pp.54-62, 24-28 June 2001.

[Opthof 1997] Opthof, T. (1997) *Sense and nonsense about the impact factor*, Cardiovascular Research, 33:1, pp. 1-7, January 1997.

[Page *et al.* 1998] Page, L., Brin, S., Motwani, R., Winograd, T. (1999) *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford Digital Library Technologies Project, 29 January 1998.

[Pajek] Networks/Pajek: Program for Large Network Analysis. http://vlado.fmf.uni-lj.si/pub/networks/pajek/

[Pellet] Pellet: open-source OWL-DL reasoner. http://pellet.owldl.com/

[Petinot *et al.* 2004] Petinot, Y., Giles, C. L., Bhatnagar, V., Teregowda, P. B., Han, H., Councill, I. G. (2004) *A Service-Oriented Architecture for Digital Libraries*, In proceedings of International Conference on Service-Oriented Computing, New York, USA, 15-18 November 2004.

[Petricek *et al.* 2005] Petricek, V., Cox, I. J., Han, H., Councill, I. G., Giles C. L. (2005) *A Comparison of On-line Computer Science Citation Databases*, In proceedings of 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2005), Vienna, Austria, 18-23 September 2005.

[Pinski & Narin 1976]

Pinski, G., Narin, F. (1976) *Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics*, Information Processing and Management, 12:5, pp. 297-312, 1976.

[Prud'hommeaux & Seaborne 2006]

Prud'hommeaux, E., Seaborne, A. (Eds) (2006) *SPARQL Query Language for RDF*, W3C. http://www.w3.org/TR/rdf-sparql-query/

[Rexa] Computer Science digital library. http://rexa.info/

[Rousseau 2006] Rousseau, R. (2006) *A case study: evolution of JASIS' Hirsch index*, Science Focus, 1:1, pp. 16-17, January 2006.

[schraefel *et al.* 2004] schraefel, m.c., Shadbolt, N.R., Gibbins, N.M., Harris, S.W., Glaser, H. (2004) *CS AKTive Space: Representing Computer Science in the Semantic Web*, In proceedings of the 13th international conference on World Wide Web, New York, USA, pp. 384-392, 17-22 May 2004.

[Schvaneveldt 1989] Schvaneveldt, R.W., Durso, F.T., Dearholt, D.W. (1989) *Network structures in proximity data* In G. Bower (Ed.), The psychology of learning and motivation: Advances in research and theory, 24, pp. 249-284, New York: Academic Press.

[Schwartzbach's H-Number Calculator]
http://www.brics.dk/~mis/hnumber.html

[Seglen 1992] Seglen, P.O. (1992) *The skewness of science*, Journal of the American Society for Information Science, 43, pp. 628-38, October 1992.

[Seglen 1994] Seglen, P.O. (1994) *Causal relationship between article citedness and journal impact*, Journal of the American Society for Information Science, 45, pp. 1-11, January 1994.

[Seglen 1997] Seglen, P.O. (1997), *Why the impact factor of journals should not be used for evaluating research*, British Medical Journal, 314, p. 498-502, 1997.

[Shadbolt *et al.* 2006] Shadbolt, N., Brody, T., Carr, L. and Harnad, S. (2006) *The Open Research Web: A Preview of the Optimal and the Inevitable*, in Jacobs, N., Eds. *Open Access: Key Strategic, Technical and Economic Aspects*, chapter 21, Oxford: Chandos.

[Shadbolt *et al.* 2006a] Shadbolt, N., Berners-Lee, T. and Hall, W. (2006) *The Semantic Web Revisited*, IEEE Intelligent Systems 21:3, pp. 96-101, January 2006.

[Shadish *et al.* 1995] Shadish, W., Tolliver, D., Gray, M., Gupta, S. (1995) *Author Judgements about works they cite: Three studies from psychology journals*, Social Studies of Science, 25, pp. 477-497, 1 August 1995.

[Skupin 2004] Skupin, A. (2004) *The world of geography: Visualizing a knowledge domain with cartographic means*, Proceedings of the National Academy of Sciences, 101 (Suppl. 1), pp. 5274-5278, April 2004.

[Smith & Eysenck 2002]
Smith, A. T., Eysenck, M. (2002) *The correlation between RAE ratings and citation counts in psychology*, Technical Report, Psychology, University of London, Royal Holloway, June 2002.

[Sombatsompop *et al.* 2004]

Sombatsompop, N., Markpin, T. and Premkamolnetr, N. (2004), *Making an equality of ISI impact factors for different subject fields*, Scientometrics, 60:2, pp. 217-235, June 2004.

[UCINET] UCINET 6: Social Network Analysis Software. http://www.analytictech.com/ucinet/ucinet.htm

[Van Rijsbergen 1979]

van Rijsbergen, C. J. (1979) *Information Retrieval* London: Butterworth.

[Weale *et al.* 2004] Weale, A.R., Bailey, M. & Lear, P.A. (2004) *The level of non-citation of articles within a journal as a measure of quality: a comparison to the impact factor*, BMC Medical Research Methodology, 4:14, 28 May 2004.

[Wolfram 2003] Wolfram, D. (2003) *Applied informetrics for information retrieval research*, Westport, CT, USA: Libraries Unlimited.

[Zhao 2005] Zhao, D. (2005) *Challenges of scholarly publications on the Web to the evaluation of science - A comparison of author visibility on the Web and in print journals*, Information Processing and Management, 41:6, pp. 1403-1418, December 2005.

Index

A

Acknowledgement Analysis 17

ACM Digital Library 21

AKTiveAuthor.................................. 35

B

Betweenness Centrality 15

C

Citation Analysis 7

  Relative 14

Citation Half-Life 15

Citebase 24

Citeseer 9

Citespace.. 15

D

Data Mediation 49

Digital Libraries.................................. 20

  Academic Paper Stores 21

  Crawlers 21

  Institutional Archives 20

Download Analysis............................ 16

DSpace 20

Dublin Core 24

E

EPrints 20

G

G-Index 18

Google Scholar 21

H

H-Index 17

Hubs and Authorities.......................... 14

Hypothesis

  Central hypothesis 2

  First sub-hypothesis 2

  Second sub-hypothesis 2

  Third sub-hypothesis 2

J

Journal Impact Factor 8

  Criticisms 10

JUNG.. 33

M

MSN Academic Live.......................... 21

O

Ontologies..26

OntologiesAKT Reference Ontology 28

Open Archive Initiative 23

OWL 26

P

PageRank..14

R

RDF 25

References 137

Reviewed Digital Library Systems..122

Rexa...19

S

Semantic Web..................................25

Semiometrics.................................. 49

- Avoided Metrics 73
- List-similarity 79
- Metadata Integration 97
- Metrics.. 69
- Ordered Lists Client 58
- Semioviewer Client Applet 59
- Technical Overview 130
- Web Services................................ 56

SimMetrics 104

SPARQL 27

T

Thesis

- Contribution 5
- Structure 4