

Distributed Systems Lab

Related Work Search Tool

Andrea Francke
afrancke@student.ethz.ch

Dorian Kind
dorian@student.ethz.ch

Prof. Dr. Roger Wattenhofer
Distributed Computing Group

Betreuer: Olga Goussevskaia, Michael Kuhn, Yvonne Anne Oswald

Zusammenfassung

Wir stellen ein Tool vor, das Methoden der Bibliometrie einsetzt, um Ähnlichkeitsbeziehungen zwischen Forschungspublikationen zu visualisieren. Als Datengrundlage wurde die *DBLP* mit Daten aus der *ACM Digital Library* erweitert. Die Applikation ermöglicht ein schrittweises Erforschen des resultierenden Graphen.

Neben unseren Betreuern hat dieser Report vor allem Personen zum Zielpublikum, die auf der Grundlage unserer gesammelten Daten und/oder unseres Tools weiterarbeiten möchten.

Inhaltsverzeichnis

1. Einführung	4
2. Grundlagen	6
2.1. Co-Citation Coupling und Bibliographic Coupling	6
2.2. Nachbarschaft über gemeinsame Autoren und Erkennen von Selbstzitierungen	8
2.3. “White Box”-Konzept	8
2.4. Einige Eigenschaften der Daten	9
2.5. JUNG	10
2.6. tagsoup	14
2.7. Bedienungsanleitung	14
2.8. Hinweise für Entwickler	18
3. Themenverwandte Arbeiten	21
4. Weiterführende Arbeiten	23
5. Erfahrungsberichte	26
5.1. Erfahrungsbericht Andrea	26
5.2. Erfahrungsbericht Dorian	28
6. Schlusswort	30
A. Appendix	32
A.1. Datenbank	32
A.1.1. tblbc	32
A.1.2. tblccc	33
A.1.3. tblca	34
A.2. Einige exemplarische Datentabellen	34
A.2.1. Die Konferenzen mit den meisten Zitierungen pro er- fasstes Paper	34
A.2.2. Co-Citation Coupling Werteverteilung	36
A.2.3. Bibliographic Coupling Werteverteilung	38

1. Einführung

Der Ausgangspunkt unseres Labs war eine Fragestellung, die unserer Meinung nach hochinteressant war - und sehr offen: Ausgeschrieben war unser Lab unter dem Titel "Community Dynamics in Social Networks". Als soziales Netzwerk, das wir dabei betrachten sollten, war die DBLP¹, eine Forschungsdatenbank, die im XML-Format frei verfügbar ist, vorgesehen. Dieses Netzwerk sollten wir genauer unter die Lupe nehmen und/oder uns eine interessante Anwendung ausdenken, die auf den Informationen basierte, die wir der Datenbank entnehmen konnten.

Die DBLP umfasst Publikations-, Konferenz- und Autorendaten von mehr als 900'000 Publikationen aus dem Bereich der Informatikforschung, jedoch keine Zitierungen und keine Abstracts. Da fast alle spannenden Anwendungen, die wir uns vorstellen konnten, auf Zitierungsdaten basierten, war für uns relativ schnell - noch ohne beschlossen zu haben, welche der Ideen wir dann mit den Zitierungsdaten verwirklichen würden - klar, dass wir versuchen wollten, angelehnt an die Idee eines Mash-Ups, die DBLP-Daten mit Zitierungs- und, wenn möglich, Abstract-Daten aus anderen Quellen anzureichern.

Kandidaten für Zweit-Datenquellen neben der DBLP gab es viele, nur sehr wenige waren jedoch für unsere Zwecke geeignet. Bei CiteSeer² fehlten die aktuellen Daten; die Daten von ISI Web Of Science³ hielten wir für zu selektiv und damit ungeeignet ([11]); Die Zitierungsdaten von Google Scholar⁴ waren weder auf dem offiziellen Wege erhältlich noch crawlbar. Eine geeignete Datenquelle fanden wir hingegen in der *ACM Digital Library*⁵, die für einen Grossteil der dort erfassten Artikel eine Webseite mit Abstract und Zitierungsliste liefert, wobei ein Teil der Zitierungen sogar als Link verfügbar ist.

¹<http://www.informatik.uni-trier.de/~ley/db/>

²<http://citeseer.ist.psu.edu/>

³<http://scientific.thomson.com/products/wos/>

⁴<http://scholar.google.com/>

⁵<http://portal.acm.org/dl.cfm>

Wir versuchten, von *ACM* entweder direkt Zitierungsdaten, oder, falls das nicht möglich wäre, die Erlaubnis zu erhalten, die *ACM Digital Library* Website zu crawlen. Die einzige Reaktion, die wir von *ACM* erhielten, war eine Antwort auf unsere zweite Anfrage, die besagte, dass wir innert Wochenfrist Bescheid bekommen würden. Da wir bis heute nichts mehr von *ACM* gehört haben, begannen wir, nachdem wir lange auf eine weitere Nachricht von *ACM* gewartet hatten, mittels in der DBLP gespeicherten Links zu *ACM*-Papern die Digital Library zu crawlen. Auf diese Weise speicherten wir rund 80'000 Seiten zu einzelnen Publikationen. Diese Seiten konnten wir mit Hilfe des Tagsoup Frameworks⁶ parsen und so Abstracts und Zitierungslisten, wo inbegriffen in den Webseiten, in unserer Datenbank speichern.

Während unser Script noch mit Crawlen beschäftigt war, galt es nun, sich unter vielen interessanten Anwendungen für eine zu entscheiden. Das Projekt unserer Wahl war ein "Related Work Search Tool"⁷. Wir wollten die Struktur des sozialen Netzwerkes dazu nutzen, Ähnlichkeiten zwischen Publikationen abschätzen und für ein gegebenes Paper oder sogar Thema verwandte Artikel ausgeben zu können. Im Idealfall würde unser Tool Forschenden, die nach Arbeiten zu einem bestimmten Thema oder anhand eines gegebenen Papers weitere Lektüre suchen und die mit Google Scholar und ähnlichen Tools unzufrieden sind, die Arbeit erleichtern.

Das Resultat unserer Arbeit ist nun ein solches Tool, das die Nachbarschaft eines Papers x als sämtliche Artikel definiert, die von x zitiert werden, die x zitieren, die von einem gleichen Paper zitiert werden wie x oder die gleiche Paper zitieren wie x . Ähnlichkeiten, auf die sich von solchen Nachbarschaften schliessen lassen, werden aber nicht zu einem Ranking ähnlicher Paper verrechnet, sondern als Struktur angezeigt. Unser Tool ermöglicht die Suche nach Schlüsselwörtern in allen gespeicherten Publikationstiteln und Abstracts. Eine wählbare Anzahl Resultate wird nicht wie üblich nur als Liste, sondern als Graph in der Ebene dargestellt, der alle Nachbarschaftszusammenhänge zwischen und die lokale und globale Relevanz von Publikationen darstellt.

⁶<http://ccil.org/~cowan/XML/tagsoup/>

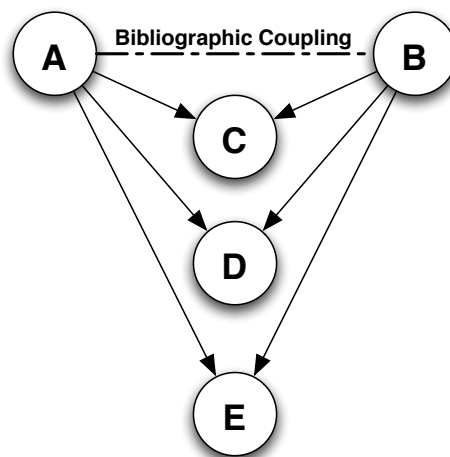
⁷Notizen zu weiteren Ideen finden sich im Abschnitt "Weiterführende Arbeiten"

2. Grundlagen

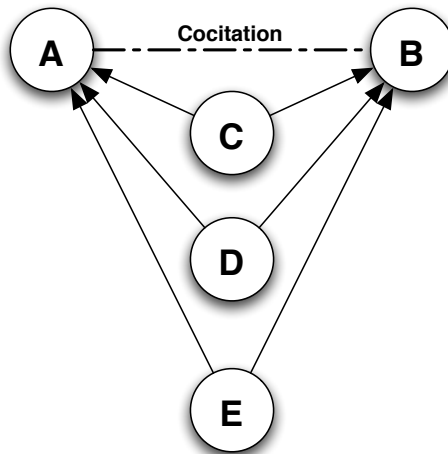
Dieser Abschnitt beleuchtet die Ideen, Methoden und Technologien, die unserem Tool zugrunde liegen, das im Handbuch-Kapitel genauer erläutert wird.

2.1. Co-Citation Coupling und Bibliographic Coupling

Was wir im letzten Abschnitt neben direkten Zitierungen als Nachbarschaftszusammenhänge von Publikationen beschrieben haben, sind zwei etablierte Konzepte der Bibliometrie und der Zitationsanalyse: Co-Citation Coupling (CCC) und Bibliographic Coupling (BC).



Bibliographic Coupling verbindet zwei Publikationen, die die selben Drittpublikationen zitieren. Das Konzept dahinter postuliert, dass zwei Publikationen umso ähnlicher sind, je mehr Publikationen sie gemeinsam zitieren (siehe z.B. [4]). Ein einfaches Ähnlichkeitsmass ist es also, für jedes Publikationspaar die Anzahl gemeinsam zitierten Drittpublikationen zu zählen.



In der “gespiegelten” Version des Bibliographic Coupling verbindet Co-Citation Coupling zwei Publikationen, die beide von der selben Drittpublikation zitiert werden (siehe z.B. [4]). Auch hier lässt sich also ein einfaches Ähnlichkeitsmass konstruieren, indem für jedes Publikationspaar die Anzahl Drittpublikationen gezählt wird, von denen beide zitiert werden.

Sowohl Bibliographic Couplings wie auch Co-Citation Couplings werden in unserem Tool als ungerichtete Kanten dargestellt. Während wir ursprünglich planten, beide zu einem Mass zu kombinieren, z.B. indem die beiden Werte addiert werden, sind nun beide Werte als separate Kanten aufgeführt. Dies aus drei Gründen: die beiden Masse repräsentieren unterschiedliche Arten von Ähnlichkeit und nehmen verschiedene Wertebereiche an: Co-Citation Couplings erreichen typischerweise höhere Werte als Bibliographic Couplings (siehe auch Appendix A.2). Dann bot sich gar nicht so viel Möglichkeit für eine Kombination der beiden Werte: Während wir für 1.5 Millionen Publikationspaare positive Bibliographic Coupling-Werte haben und für knapp 700'000 positive Co-Citation Coupling-Werte, sind nur bei knapp 78'000 Publikationspaaren beide Werte positiv. Nur bei einem kleinen Anteil der Publikationspaare wird also der Graph durch zwei Coupling-Kanten unübersichtlicher gemacht, der Vorteil der separaten Repräsentation der beiden Kanten überwiegt unserer Meinung nach also. Der dritte Grund wird im Abschnitt “White Box'-Konzept” erläutert.

Anmerkungen dazu, wie die beiden Werte zusammengezählt werden könnten und eine Kritik am möglicherweise zu einfachen Aufsummieren der Couplings finden sich im Abschnitt “weiterführende Arbeiten”.

Die DBLP enthält sehr viele Publikationen mit identischem Titel. Alleine der Titel “Home Page” taucht 10’060 mal auf, “Editorial” liegt auf dem zweiten Platz mit 1077 Publikationen. Betrachtet man nur Publikationen, die in unseren Daten mindestens einmal zitiert werden, reduzieren sich diese Zahlen massiv: Nur zwölf Titel tauchen mehr als zweimal auf, hingegen gibt es 206 Titel, die genau zweimal auftauchen. In vielen Fällen sind dies Paper, die zweimal veröffentlicht wurden: einmal an einer Konferenz und einmal in einem Journal. Da diese Publikationen natürlich durchs Band hohe Bibliographic Coupling-Werte haben, haben wir Publikationspaare mit identischem Titel bei der Bestimmung von Bibliographic Coupling- und Co-Citation Coupling-Werten herausgefiltert.

2.2. Nachbarschaft über gemeinsame Autoren und Erkennen von Selbstzitationen

Neben den Zitierungen und Couplings gibt es ein weiteres Ähnlichkeitsmass in unserem Tool: Der Anteil gemeinsamer Autoren zweier Papers. Dieser wird für jedes Publikationspaar gezählt und mit der kleineren Autorenzahl der beiden Publikationen skaliert und nimmt somit einen Wert zwischen 0 (keine Kante) und 1 an. Die “Formel” dazu ist im Handbuch-Abschnitt aufgeführt. Wir hatten zwei Motivationen, dieses Ähnlichkeitsmass in unser Tool aufzunehmen: Einerseits halten wir gleiche Autoren bei verschiedenen Publikationen für einen wichtigen Indikator für thematische Ähnlichkeit. Andererseits hilft dieses Mass, Selbstzitationen zu indentifizieren, die schwierig binär zu definieren sind: Ab wievielen gemeinsamen Autoren ist eine Zitierung eine Selbstzitierung? Unser Mass liefert einen graduellen Indikator für die Wahrscheinlichkeit, dass es sich bei einer Zitierung um eine solche handelt.

2.3. “White Box”-Konzept

Mit steigenden Datenmengen, über die man sich ungleich wie früher in papierenen Archiven keinen Überblick mehr verschaffen kann ohne ein Information Retrieval-Werkzeug zu verwenden, wird die Datensuche immer mehr zum Laborexperiment. Wer experimentiert, sollte aber die Implikationen und die ungefähre Funktionsweise seiner Instrumente kennen. Dass sich aber

kaum ein Benutzer Zeit nimmt, seine Suchmaschine zu verstehen, ist eine grosse Schwierigkeit im Information Retrieval.

Diese These ist eine Motivation für die Graphstruktur unseres Tools: Wir wollen versuchen, durch das Mapping der Verwandtschaften einer Publikation mit anderen Publikationen die Funktionsweise unseres Tools offenzulegen, ohne dass sich der Benutzer dafür gross einlesen muss. Die Struktur, die wir zum Information Retrieval benützen, legen wir grösstenteils frei. Was nicht freigelegt wird und ein Blackbox-Verfahren bleibt, ist die Relevanzbewertung der Publikationen im Tool via zugehörige Konferenz oder zugehöriges Journal (siehe Handbuch-Kapitel).

2.4. Einige Eigenschaften der Daten

Gesammelt wurden die Daten mit einem Script, welches die von *DBLP* zur Verfügung gestellte XML-Datei mit sämtlichen Einträgen nach Publikationen durchsucht, welche auf der *ACM*-Website verfügbar sind. Die entsprechenden HTML-Seiten wurden heruntergeladen und mit *tagsoup* nach Zusammenfassung und Zitierungsliste geparst, wobei die gewonnenen Informationen in eine MySQL-Datenbank abgespeichert wurden. Als Identifikation der einzelnen Artikel dient hierbei auf *ACM*-Seite die sogenannte DOI oder “Digital Object Identifier”¹, in der Datenbank wurde ein eigener Key verwendet, sodass ein Mapping zwischen DOI und Datenbank-Keys nötig ist (die Tabelle *tblpubDOI*). Von den geparsten Zitierungen konnten durchschnittlich circa fünf pro Paper aufgelöst werden. Ausgehende Zitierungen fanden wir von rund 52'000, eingehende Zitierungen für knapp 63'000 Artikel; insgesamt konnten wir Informationen über fast 240'000 Zitierungen speichern.

Unsere Hauptdatenquelle ist folglich der Graph, der aus allen gefundenen Zitierungen innerhalb der von uns untersuchten Publikationen gebildet wird. Wie bereits erwähnt, umfasst er etwa 85'000 Knoten sowie annähernd 240'000 Kanten. Der durchschnittliche Grad von 2.74 ist damit relativ niedrig, was uns befürchten liess, dass der Graph zu segmentiert sein könnte, um damit arbeiten zu können. Bei genauerer Analyse stellte sich heraus, dass die grösste Komponente mehr als 93% aller Knoten umfasste und die restlichen 2328 Komponenten alle kleiner als 15 Knoten waren (Durchschnitt 2.51). Die Verbundenheit scheint also kein Problem zu sein, und der Durchmesser des

¹Siehe www.doi.org für weitergehende Informationen zu DOI

Graphen dürfte vergleichsweise gering sein. Weiterhin untersuchten wir die Gradverteilung für eingehende und ausgehende Kanten. Wie Abbildungen 2.1 und 2.2 zeigen, lassen sich beide Verteilungen gut durch ein Potenzgesetz approximieren, ein Indiz für ein skalenfreies Netzwerk. Diese Beobachtungen deuten darauf hin, dass der Zitierungsgraph ein *small world network* bildet, was aufgrund seiner Entstehungsweise nicht weiter überrascht.

Interessanterweise fanden sich einige Zyklen im Zitierungsgraph, und – noch überraschender – auch einige Fälle, wo ein Paper sich selbst zitierte. Letzteres dürfte meist auf zwei Artikel mit gleichem Namen zurückzuführen sein²; es gibt allerdings effektiv Publikationen, die sich selbst zitieren³. Über die Gründe dafür kann nur spekuliert werden. . .

2.5. JUNG

*JUNG*⁴ ist eine java-Library, die Funktionalität zur Modellierung, Analyse und Darstellung von Daten bereitstellt, die als Graph repräsentiert werden können. Wir haben *JUNG* extensiv im Verlaufe der Implementation des Related Work Search Tools benützt, um eine interne Darstellung der betrachteten Graphen zu ermöglichen sowie zur Visualisierung derselben. *JUNG* ist modular und generisch aufgebaut und lässt sich verhältnismässig einfach an die gegebenen Anforderungen anpassen. Die Hauptkonzepte von *JUNG* sind die folgenden:

- Ein *Graph* ist die grundlegende Datenstruktur in *JUNG*. Ein *Graph* kann undirected, directed oder generisch sein, was sich jeweils auf die akzeptierte Art Knoten und Kanten auswirkt. Graphen können parallele Kanten erlauben oder verbieten. *Graph* verfügt über Funktionalität zum Hinzufügen und Entfernen von Knoten und Kanten sowie den Zugriff auf dieselben.
- Ein *Vertex* stellt einen Knoten dar. Auch er existiert in den Versionen undirected, directed oder generisch, je nach dem, mittels was für Typen von Kanten er verknüpft werden kann. Methoden zum Zugriff

²Beispielsweise existieren vier Publikationen in der DBLP mit dem Titel “Software For Simulation”, alle vom selben Autoren und in vier aufeinander folgenden Jahren an der “Winter Simulation Conference” vorgestellt.

³Als Beispiel mag <http://portal.acm.org/citation.cfm?id=320546> herangezogen werden, wo die vierte Zitierung auf dieselbe Publikation verweist. . .

⁴jung.sf.net

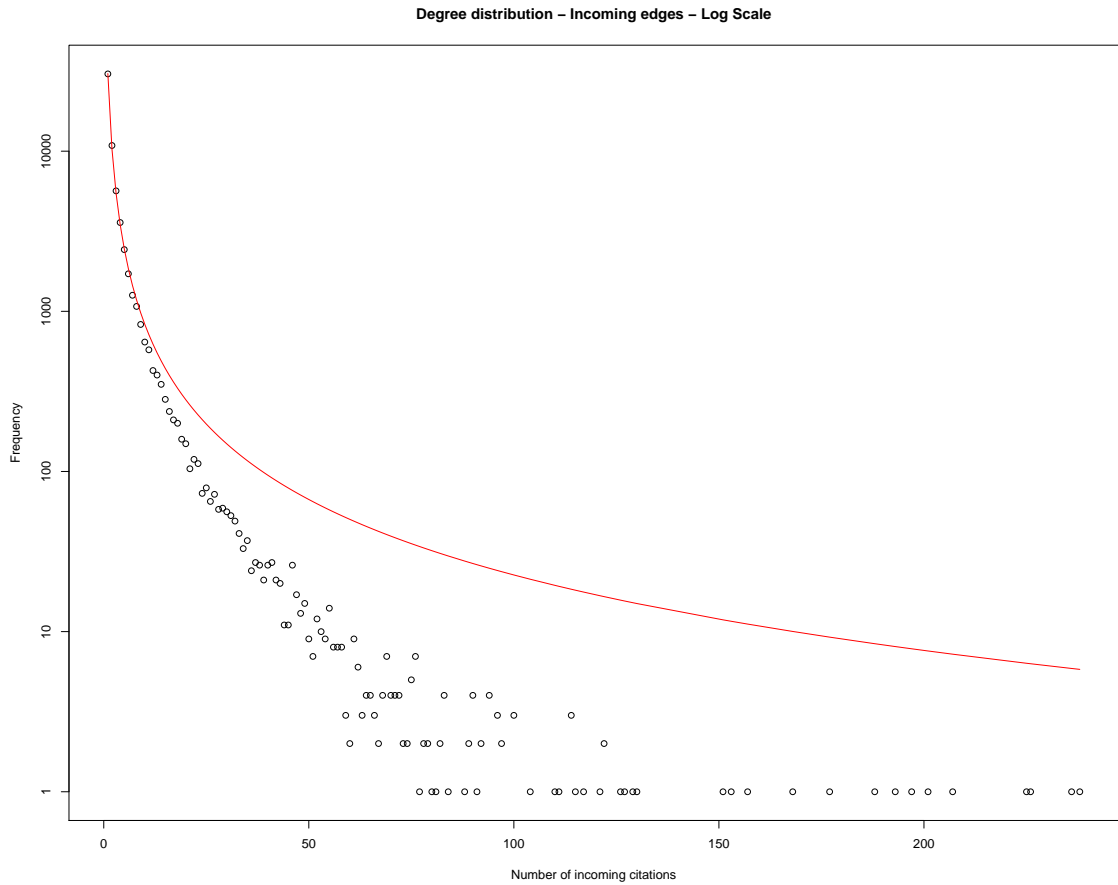


Abbildung 2.1: Plot der Eingangsgradverteilung (logarithmierte y-Achse). Die rote Linie markiert einen least squares non-linear fit mit $y = k_0 * x^{k_1}$ für $k_0 = 30'533.24$, $k_1 = -1.57$

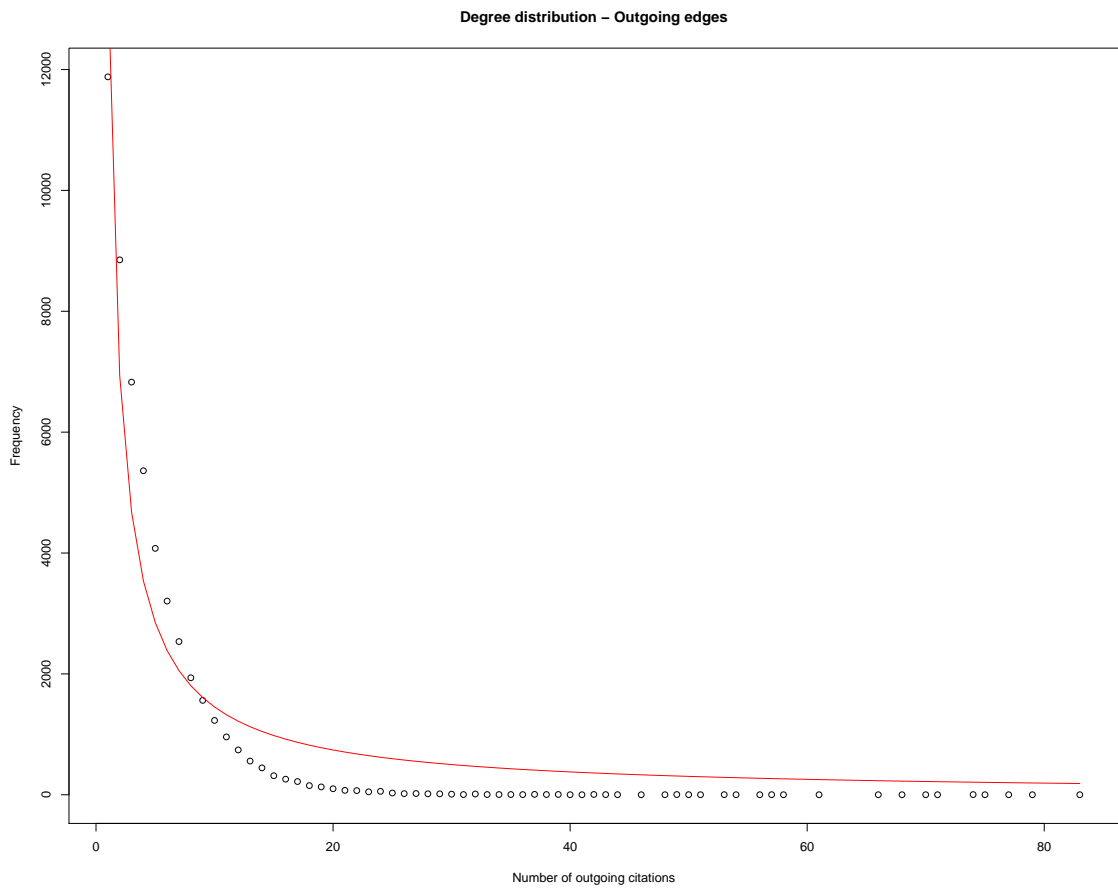


Abbildung 2.2: Plot der Ausgangsgradverteilung. Die rote Linie markiert einen least squares non-linear fit mit $y = k_2 * x^{k_1}$ für $k_2 = 13'600.24$, $k_1 = -0.97$

auf ein- und ausgehende Kanten und den beinhaltenden Graphen werden zur Verfügung gestellt, sowie zum Kopieren von Knoten in andere Graphen, wobei die Identitätseigenschaft *isEqual()* und Annotationen erhalten bleiben. Der letzte Punkt ist insbesondere wichtig, um Teilgraphen von bestehenden Graphen effizient abzubilden.

- Eine *Edge* ist eine Kante, die immer entweder *directed* oder *undirected* sein muss. *Edges* werden bereits im Konstruktor an zwei Knoten gebunden, so dass keine verwaisten Kanten entstehen sollten. Auch hier existiert wieder Funktionalität zum Zugriff auf Graph und anliegende Knoten sowie zum Kopieren einer Kante in andere Graphen.

Wie bereits angetönt, bietet *JUNG* eine simple, aber mächtige Möglichkeit zum Annotieren von beliebigen Graphenelementen mittels des Interfaces *User-DataContainer*, welches unter Verwendung von Schlüsselobjekten das Zuordnen von beliebigen Objekten zu Elementen eines Graphen ermöglicht. Da die entsprechenden Daten in einer separaten Datenstruktur gespeichert werden, bietet diese Vorgehensweise erhebliche Vorteile gegenüber eines naheliegenden eigenen Erweiterns der Grundklassen:

- Wenn nur wenige Elemente eines Graphen annotiert werden müssen, wird gegenüber dem Erweitern von *Vertex* oder *Node* Speicher gespart, da nur soviel Daten abgelegt werden wie tatsächlich nötig. In unserem Fall verfügt der komplette Graph beispielsweise über circa 240'000 Kanten und 80'000 Knoten, wovon zu jedem Zeitpunkt jedoch jeweils nur einige Dutzend über Annotationen in der Grössenordnung von einigen *Integers* und *Strings* verfügen, d.h. zwischen wenigen Bytes bis zu um die 200 Bytes. Das ergibt bereits eine nicht zu verachtende Reduktion des Speicherbedarfs.
- Mehrere Elemente können sich ohne weiteres Dazutun Annotationen teilen, was in unserem Fall praktisch zum Kopieren von Knoten aus dem gesamten Zitierungsgraphen in den darzustellenden Subgraphen ist.

Weiterhin bietet *JUNG* eine Auswahl von relevanten Algorithmen wie Shortest Path, HITS, Betweenness und ähnlichem. Für uns von Bedeutung war hierbei nur die Analyse der Gradverteilung, der *WeakComponent-Clusterer* zur Abschätzung der Konnektivität unseres Graphen und der *k-Nachbarschaftsfilter* zum Darstellen der namensgebenden *Neighborhood* einer Publikation.

Der dritte wichtige Teil von *JUNG* besteht in der Visualisierung von Graphen. Grundsätzlich wird unterschieden zwischen dem Layout, welches die Positionen der Knoten des Graphen definiert, und dem Renderer, der gemäss diesen Vorgaben die Elemente des Graphen zeichnet. Beide Klassen respektive Interfaces sind weitreichend konfigurierbar. Für einen möglichst schnellen Start wird eine Klasse *VisualizationViewer* zur Verfügung gestellt, welche die beiden Konzepte beinhaltet und von *JPanel* erbt, so dass sie direkt in eine Swing-basierte Anwendung eingebunden werden kann. Das Related Work Search Tool verwendet entweder *SpringLayout* (ein *spring embedder*) oder *FRLayout* (verwendet den Fruchterman-Reingold Algorithmus) und einen für massgeschneiderte Darstellung ausgiebig erweiterten *PluggableRenderer*.

2.6. tagsoup

Das Motto von *tagsoup* ist “Just Keep On Truckin”, und genau das tut es auch: *tagsoup* parst HTML, ganz egal ob von Tim Berners-Lee persönlich geschrieben oder aus der HTML-Hölle namens *Microsoft FrontPage* stammend. *tagsoup* ist ein kleines java-Framework, dass mittels standardisierter SAX-Anbindungen zum programmatischen Auslesen von HTML gebraucht werden kann. In Falle des Related Work Search Tools benötigte das Einbinden von *tagsoup* weniger als 10 Zeilen Code. Es wird ein SAX-Handler übergeben, dessen Callback-Methoden für den Start und das Ende jedes angetroffene Elementes aufgerufen werden. Mittels einer Statemachine kann so mit relative geringem Aufwand der Inhalt eines bestimmten interessierenden Elementes ausgelesen werden, in unserem Fall das *paragraph*-Element, welches das Abstract enthält sowie die *table* mit der Zitierungsliste.

2.7. Bedienungsanleitung

Die Benutzung des Related Work Search Tool gestaltet sich denkbar einfach und sollte eigentlich ziemlich intuitiv sein. Hier dennoch einige Hinweise.

Abbildung 2.3 zeigt die Benutzeroberfläche. Sie ist in zwei Fenster aufgeteilt, zur linken ist der “Graph Display” erkennbar, auf der rechten Seite befindet sich das “Search”-Fenster, welches die gerade angezeigten Publikationen auflistet und weiterreichende Informationen zum momentan ausgewählten Artikel bietet.

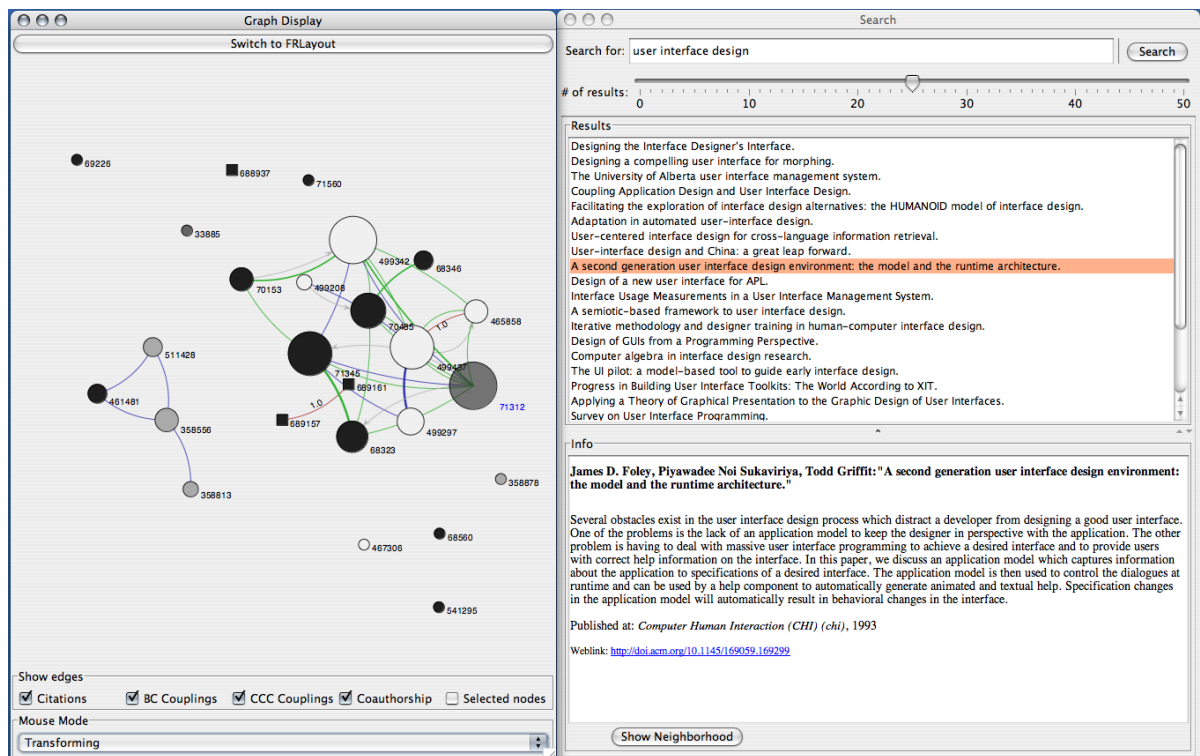


Abbildung 2.3: Die Benutzeroberfläche des Related Work Search Tools

Eine neue Suche wird eingeleitet, in dem die entsprechenden Suchbegriffe in das mit "Search For" beschriftete Feld eingegeben werden und *Return* gedrückt oder alternativ auf "Search" geklickt wird. Mit dem Regler "# Of Results" kann die Menge der angezeigten Treffer eingeschränkt werden. Gesucht wird in den Zusammenfassungen sowie den Titeln der Publikationen. Die Suchbegriffe können mit Operatoren kombiniert werden, so sucht "**Ein beispielhafter Suchtext**" (man beachte die Anführungszeichen) nach genau diesem Ausdruck; während "**wireless network -sensor**" nach Artikeln sucht, welche die ersten beiden Suchbegriffe enthalten, nicht aber den dritten.

Die gefundenen Artikel werden nach Relevanz geordnet im mittleren Teil des Suchfensters angezeigt. Bei Auswahl einer Publikation werden im unteren Teil Details dazu angezeigt:

- Autoren
- Titel
- Zusammenfassung ("Abstract")
- Ort und Jahr der Publikation
- Weblink

Möglicherweise sind eine oder mehrere Detailangaben für einen bestimmten Artikel nicht verfügbar. Auf den Knopf "Show Neighborhood" wird zu einem späteren Zeitpunkt noch eingegangen.

Die im Verlaufe einer Suche gefundenen Artikel werden im Graphenfenster als Knoten angeordnet, und mit verschiedenartigen Kanten verknüpft. Dabei kommen folgende Gestaltungselemente zum Einsatz:

- Die *Grösse* eines Knoten repräsentiert seine relative Verknüpfung mit dem Rest des Graphen. Je grösser der Knoten, desto besser ist er mit anderen Knoten verknüpft.
- Die *Form* eines Knoten unterscheidet Artikel, die an Konferenzen vorgestellt wurden und eine runde Form aufweisen, von solchen, die in Journals publiziert wurden (quadratische Form). Eine Besonderheit stellt die Sternform dar, welche später noch erwähnt wird.

- Die *Helligkeit* eines Knoten stellt das Ranking der Konferenz/des Journals dar, wo er publiziert wurde⁵. Die Skala reicht hierbei von 1 (sehr wichtige Konferenz, dunkelste Färbung) bis 3 (weniger wichtige Konferenz, eher helle Färbung), wobei 0 eine unbedeutende oder unbekannte Konferenz bedeutet, was durch eine sehr helle Färbung dargestellt wird.

So weit zu den Knoten. Die Kanten, welche sie verknüpfen, haben ebenfalls bestimmte graphische Merkmale:

- *Graue*, gerichtete Kanten stellen Zitierungen dar.
- *Grüne Kanten* bedeuten, dass eine *bibliographic coupling* Beziehung zwischen den beiden Artikeln besteht, hierbei wird die Stärke der Verbundenheit, d.h. die Anzahl von Drittartikeln, welche von beiden Publikationen zitiert werden, durch die Breite der Kante dargestellt.
- *Blaue* Kanten repräsentieren Co-Zitationen (co-citations), wobei auch hier wieder die Breite der Kante die Stärke der Verbundenheit anzeigt.
- *Rote* Kanten schliesslich verbinden zwei Knoten, deren entsprechende Artikel gemeinsame Autoren aufweisen. Rote Kanten haben als Besonderheit ein Label, welches den Anteil gemeinsamer Autoren r angibt. Für Publikationen A und B gilt:

$$r = \frac{\text{Anzahl gemeinsamer Autoren}}{\min(\text{Anzahl Autoren von } A, \text{Anzahl Autoren von } B)}$$

Die Darstellung der unterschiedlichen Kantenarten kann mithilfe der am unteren Rand angeordneten Checkboxen individuell ein- oder ausgeschaltet werden. Mittels des Pop-up-Menüs am unteren Rand des Graphenfensters wird zwischen Transformations- und Auswahlmodus gewechselt. Im *Transformationsmodus* kann der Graph durch Drücken und Halten der Maus verschoben werden, wird hierbei zusätzlich *Shift* gedrückt, wird der Graph gedreht. Zoomen ist durch Betätigen des Mousrads in beiden Modi möglich. Der *Auswahlmodus* dient zur Selektion eines oder – durch Aufziehen eines Auswahlrechteckes – mehrerer Knoten. Ebenfalls können in diesem Modus ausgewählte Knoten verschoben werden. Die Checkbox “Selected Nodes” blendet alle Kanten aus, die nicht mit einem ausgewählten Knoten verbunden sind, womit ein Subgraph genauer untersucht werden kann.

⁵Das Rankingdaten beruhen auf [7]

Wenn eine Publikation durch Auswahl im Graphfenster oder in der List der Suchtreffer ausgewählt ist, kann durch Druck auf “Show Neighborhood” ein neuer Graph dargestellt werden, der den ausgewählten Artikel A als Wurzel hat (dargestellt durch einen Stern, siehe Abbildung 2.4) und aus den Publikationen besteht, die entweder von A zitiert werden oder ihrerseits A zitieren. So kann der Zitierungsgraph schrittweise erforscht werden.

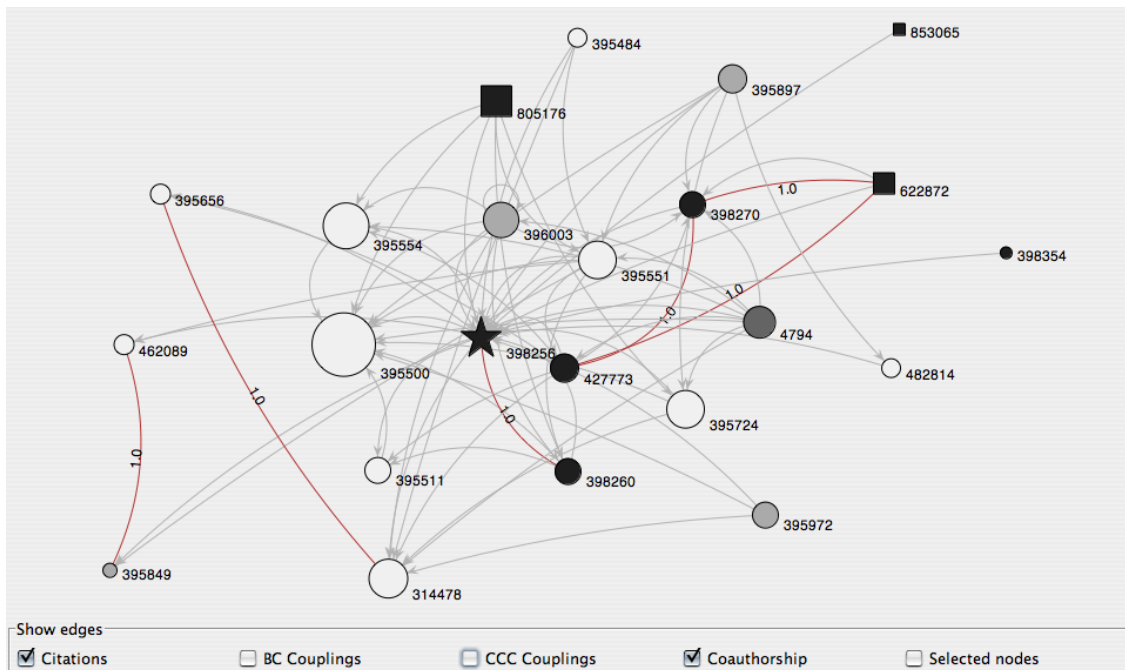


Abbildung 2.4: Die Wurzel einer Nachbarschaft wird durch einen Stern dargestellt (BC- und CCC-Kanten sind ausgeblendet)

2.8. Hinweise für Entwickler

Das Related Work Search Tool ist in Java (Version 1.5.0) geschrieben und besteht im Wesentlichen aus 3 Klassen:

`NeighborhoodVisualizer.java` ist die Hauptklasse und beinhaltet `main()`. Sie ist für das Aufsetzen der Visualisierungsumgebung zuständig, d.h. das Erstellen des GUIs. `NeighborhoodVisualizer` ist hierbei gleichzeitig View und Controller. Dazu kommen einige innere Klassen, von de-

nen die wichtigste *AuxFrame* ist, welche das “Search”-Fenster darstellt und kontrolliert. *SimpleVertexShapeFunction*, *SimpleNodeColorFunction* und *SimpleToolTipFunction* sind Helfer-Klassen, welche an *JUNG* zum Zeichnen von Knoten übergeben werden (vergleiche die *JUNG*-Klasse *PluggableRenderer*). *DirectionDisplayPredicate* dient dem Ausblenden von spezifischen Kantentypen.

NeighborhoodModel.java ist das Datenmodell. Es verwaltet über *JUNG* einen gesamten Zitierungsgraphen und den momentan dargestellten Subgraphen (die *neighborhood*). *NeighborhoodVisualizer* ruft Methoden in *NeighborhoodModel* auf, um eine neue Suche einzuleiten oder die Nachbarschaft eines ausgewählten Knotens anzuzeigen. Dazu erstellt *NeighborhoodModel* zunächst den Subgraphen und fügt dann die verschiedenen Kantentypen hinzu (mittels *addCouplingEdges()*). Die momentan angezeigten Knoten werden als *Publications* (siehe unten) gespeichert und *NeighborhoodVisualizer* für Details zur Verfügung gestellt.

NeighborhoodDB.java stellt die Schnittstelle zur Datenbank dar. Alle relevanten Queries werden als *PreparedStatements* initialisiert und dann entsprechend aufgerufen. Daraus folgt auch, dass für das Benutzen des Related Work Search Tool eine MySQL Datenbank vonnöten ist, welche die relevanten Tabellen *tblabstracts*, *tblauthors*, *tblbc*, *tblca*, *tblccc*, *tblcitationedges*, *tblplaces*, *tblpub* und *tblpubauthors* aufweisen muss. Siehe dazu auch Appendix A.1.

Weitere Klassen sind *Publication*, welche die Details einer Publikation abbildet (Titel, Zusammenfassung, Autoren etc.) und *NeighborhoodConstants.java*, worin – *nomen est omen* – einige Konstanten definiert werden.

Als build-tool wird *ant*⁶ in Verbindung mit *one-jar*⁷ verwendet, wodurch alle Klassen sowie die benötigten Libraries *JUNG*⁸, *colt*⁹, *mysql-connector-java*¹⁰, *commons-lang*¹¹ und *commons-collections*¹² in ein einziges jar-File gepackt werden können, um so dem Benutzer das Herunterladen

⁶<http://ant.apache.org>

⁷<http://sourceforge.net/projects/one-jar>

⁸<http://jung.sourceforge.net/>

⁹<http://dsd.lbl.gov/hoschek/colt/>

¹⁰<http://www.mysql.com/products/connector/j/>

¹¹<http://jakarta.apache.org/commons/lang/>

¹²<http://jakarta.apache.org/commons/collections/>

derselben und frustrierendes Hantieren mit `$CLASSPATH` zu ersparen. *colt* und *commons-collections* sind übrigens Voraussetzung für den Betrieb von *JUNG*, die *commons-lang*-Library wird zum Unescapen der HTML-Entities in den Zusammenfassungen der Artikel benötigt.

Für das Erweitern des Codes wird insbesondere ein Einarbeiten in *JUNG*, empfohlen, im Speziellen die Packages `edu.uci.ics.jung.graph` sowie `edu.uci.ics.jung.visualization`. An dieser Stelle noch ein Wort zu den verwendeten Layout-Klassen der Visualisierung: Sowohl *SpringLayout* wie auch *FRLayout* sind iterierende Layouts, d.h. sie weisen nicht von Anfang an die optimale Plazierung der Knoten auf, sondern erst nach einigen Iterationsschritten (via *advancePositions()* oder durch den *RelaxerThread* des *VisualizationModels*). Da dies aber bei unseren Graphen aber immer sehr lange dauerte und sich das Layout teilweise gar nie stabilisierte, haben wir uns entschieden, die Iteration nicht durchzuführen, sondern mithilfe von *VisualizationViewer.suspend()* gleich die erste Version zu verwenden. Diese Vorgehensweise ist aber sicher noch verbesserungswürdig.

3. Themenverwandte Arbeiten

Neben der bereits in den vorausgehenden Abschnitten erwähnten Literatur sind hier ein paar Artikel aufgeführt und kurz kommentiert, deren Lektüre hilfreich sein könnte für jemanden, der auf der Grundlage unseres Labs weiterarbeiten möchte.

- Einen guten Überblick über mögliche Datenanalysen im Zusammenhang mit sozialen Netzwerken liefert [3]. Der Artikel beschränkt sich auf ein Datenbank-spezifisches Subset der DBLP mit ca. 30000 Papers und wendet darauf grundlegende graphen-theoretische und small-world-relevante Analysemethoden an. Wichtige Schlagwörter sind Lotka's Law, Giant Component, Jaccard-Distanz, Clustering Coefficient, Geodesics (shortest path), Centrality mittels Betweenness und Closeness.
- Während Unterlagen zur Zitationsanalyse leicht findbar sind, ist die Kritik daran an vielen Stellen weniger prominent vertreten. Da die Arbeit mit Citation Analysis aber sehr heikel sein kann, insbesondere wenn man etwas bewerten will, hier drei Quellenangaben für kritische Lektüre zur Zitierungsanalyse und ihren Grenzen. [12] bietet eine gute Auflistung von Problembereichen der Citation Analysis und einen guten Start in diesem Bereich. Weitere gute Lektüren zu diesem Thema sind [11] (Vortrag) oder [10] (Artikel) von Friedemann Mattern. Vorallem der Vortrag bietet einen schnellen Überblick.
- [5] verfolgt den gegenteiligen Ansatz unserer "white box"-Strategie der Darstellung von Suchresultaten als Graph, für den keine zwei Masse (unterschiedliche Verwandtschaftstypen, Relevanzmasse) miteinander verrechnet werden: Die Autoren zeigen, dass mittels Kernel Methods Co-Citations bzw. Bibliographic Coupling mit Kleinbergs HITS Importance zu einem parametrisierten Spektrum von Ähnlichkeits- und Relevanzmassen kombiniert werden können. Laut Paper subsummieren Kernel Methods beide Masse: Am einen Extrem des Parameterbe-

reichs erhält man die Co-Citation- bzw. Bibliographic Coupling-Werte für Ähnlichkeit, am anderen Extrem die HITS-Werte für Relevanz.

- Eine Publikation die dagegen argumentiert, Ähnlichkeitsmasse anhand der Co-Citation-Matrix zu berechnen und stattdessen vorschlägt, die Citation-Matrix zu verwenden, ist [8].
- Eine unserer Fragestellungen war, was man an Google Scholar verbessern könnte: [13] erläutert Vor- und Nachteile von Google Scholar und macht einige Verbesserungsvorschläge. Anders als unsere Kritik zielt das Paper aber nicht auf das Ranking von Google Scholar. Die Autorin teilt die Ansicht unserer Betreuer, das im Bereich “similar-document retrieval system” Raum für Verbesserungen da ist.

4. Weiterführende Arbeiten

Dieser Abschnitt umfasst eine Sammlung von Punkten, wie man unsere Arbeit erweitern und verbessern könnte, ebenso wie Ideen, was man mit den von uns gesammelten Daten unabhängig von einem Related Work Search Tool machen könnte.

- Der wichtigste und grundlegende weitere Schritt wäre es, **weitere Datenquellen** für Zitierungen zu erschliessen und zu integrieren. Namentlich sind dies IEEE¹, Springerlink² und Elsevier³. Weiter könnte es nötig sein und vollständigere Daten liefern, PDFs von Publikationen nach Zitierungen zu durchsuchen, wobei das richtige Parsen von unterschiedlichen Zitierungsstilen eine grosse Hürde darstellt.
- Die **Langsamkeit von Jung** ist ein weiteres zentrales Problem. Verbesserungen in diesem Bereich würden unser Tool sehr viel benutzbarer machen.
- Da wir unser Tool eher als “Proof Of Concept”-Arbeit sehen denn als fertiges Arbeitswerkzeug, war die Gestaltung und die Evaluation der Suchmethode - wie werden die Artikel, die als Graph dargestellt werden, gefunden und rangiert? - eher nebensächlich. Für ein arbeits-tauglicheres Tool müsste dieser Bereich genauer angeschaut werden.
- Wie kann man garantieren, dass von einem Related Work Search Tool die **aktuellsten Forschungsdaten** geliefert werden? Gerade diese seien laut unseren Betreuern am wichtigsten, aber am schwersten zu finden. Im Idealfall wären die Daten, auf denen unser Tool basiert, automatisch aktualisierbar.
- Die **Länge eines Dokumentes** mitzubersichtigen könnte akkuratere Ähnlichkeitsmasse liefern. Je länger ein Dokument, desto mehr

¹<http://ieeexplore.ieee.org>

²<http://www.springerlink.com>

³<http://www.sciencedirect.com/>

ausgehende Zitierungen; je länger ein Dokument, desto höher die Wahrscheinlichkeit, dass zwei ausgehende oder zwei einkommende Zitierungen thematisch nicht verwandt sind.

- Eine Weiterentwicklung, die wohl viel bringen würde, aber wohl schwierig zu realisieren ist, wäre, zusammen mit den Zitierungskanten **Anchortexts** zu speichern. Die Motivation dahinter ist folgende: Von einer Zitierung kann nicht unbedingt darauf geschlossen werden, dass es sich um eine positive Würdigung der Zielpublikation handelt. Gerade Fehlpublikationen wie z.B. das berühmte “Spinat enthält viel Eisen” werden teilweise sehr häufig zitiert - als negatives Beispiel.
- Wie im Grundlagenabschnitt erwähnt, hatten wir ursprünglich geplant, **CCC- und BC-Werte miteinander zu verrechnen**, dies dann aber nicht getan. Will man die beiden Werte zu einem kombinieren, ist es unserer Meinung nach wichtig zu wissen, dass Kritiker des Bibliographic Coupling der Meinung sind, dass ein solches Zitierungspaar nicht eine konstante Ähnlichkeitseinheit bildet, sondern vielmehr eine Ähnlichkeitswahrscheinlichkeit unbekannter Grösse. Die CCC- und BC-Werte zu addieren würde die Ähnlichkeit vermutlich also nicht adäquat abbilden (Quelle: [4], [9]).
In diesem Zusammenhang muss man sich natürlich auch fragen, ob es für ein Ähnlichkeitsmass angebracht ist, Co-Citations bzw. Bibliographic Couples für sich separat einfach aufzusummieren, oder ob man hier nicht besser ebenfalls einen auf Wahrscheinlichkeiten basierenden Wert nehmen würde, der das Ähnlichkeitsmass mit steigender Anzahl Co-Citations/Bibliographic Couples für ein Publikationspaar überproportional steigen lassen würde⁴. Bei einer solchen Abänderung des Ähnlichkeitsmasses müsste man allerdings auch den (gegenläufigen) Matthäuseffekt⁵ (auch: “preferential attachment”) beachten.
- Unsere am weitesten verfolgte Idee neben dem Related Work Search Tool war es, das Thema eines **Konferenzenratings** weiterbearbeiten: Was ist ein guter Ort, um ein Paper zu veröffentlichen, welche Konferenz eignet sich weniger? Konkret war hier die Idee, Michael Kuhns Paper [7] um einen Layer ergänzen. Dazu müssten aber vollständigere

⁴zum Beispiel: Sei sum_{ccc} die Anzahl Ko-Zitierungen für ein Publikationspaar, p die Ähnlichkeitswahrscheinlichkeit bei einer Zitierung; das Ähnlichkeitsmass für das Publikationspaar könnte man proportional wählen zu $1 - (1 - p)^{sum_{ccc}}$

⁵<http://de.wikipedia.org/wiki/Matth%C3%A4useffekt>

Zitierungsdaten vorhanden sein. Hier liessen sich bekannte Algorithmen wie HITS [6] auf interessante Art und Weise anpassen (“eine gute Konferenz akzeptiert ’nur’ gute Papers, während ein guter Professor an guten und an schlechten Konferenzen publiziert, ein schlechter hingegen ’nur’ an schlechten Konferenzen.” - Autoren und Konferenzen liefern also nicht die selbe Qualitätsgarantie an ihr jeweiliges Gegenüber.)

- Ein weiteres mögliches Thema wäre, **dynamische Aspekte** zu betrachten. Man könnte bekannte Algorithmen, bei denen zwischen Ausgangs- und Zieldokument des Links keine Zeitdifferenz liegen, wie z.B. HITS und PageRank, auf Zitierungen über Zeitdifferenzen hinweg anzupassen. Weitere Fragen wären: wie kann man nach Konjunkturen von gewissen Themen suchen? Wie erkennt man Trends und “Up and Coming Topics”? Wie entwickelt sich eine neue Konferenz, z.B. Mobihoc⁶?

Ein Paper, das sich anhand eines Co-Autoren-Netzwerks mit dynamischen Aspekten befasst, ist [2]. Die Datenbasis sind hier fast 300'000 Papers aus Mathematik und Neurowissenschaften von 1991 - 1998. Thema ist hier u.a. der Zusammenhang zwischen dem Grad eines Knoten und der Wahrscheinlichkeit, dass er noch weitere Kanten erhält; gewonnene Erkenntnisse werden mittels Monte-Carlo Simulation eines kleineren Graphen validiert. Schlagwörter sind hier: Scale-Free Networks, Average Separation, Clustering Coefficient, Average Degree, Preferential Attachment, Internal vs. External Links, Continuum Theory.

Mit einem ähnlichen Thema, nämlich Gruppenbildung und -evolution in sozialen Netzwerken, befasst sich [1]. Die Motivation hier ist die These, dass Innovationsdiffusion und allgemein thematische Trends und Entwicklungen eng verknüpft seien mit der Bildung und Entwicklung von Gruppen, die sich mit den entsprechenden Themen befassen.

⁶<http://www.sigmobile.org/mobihoc/>

5. Erfahrungsberichte

5.1. Erfahrungsbericht Andrea

Ich glaube, unser Lab unterscheidet sich in manchen Aspekten von einem typischen Lab. Während viele über Probleme in der Zusammenarbeit und mit der Betreuung klagen, waren gerade diese beiden Punkte für mich positive Aspekte unseres Labs. Die Zusammenarbeit mit Dorian war toll; unsere Aufgabenteilung ging sehr gut und wir ergänzten uns gut in unseren Interessen.

Am Anfang war ich erstaunt darüber, dass wir uns wöchentlich mit den Assistierenden trafen; bei meiner Semesterarbeit, für die ich parallel zum Lab arbeitete, war ich viel mehr auf mich selbst gestellt, was für mich auch zur Vorstellung von einer “selbstständigen Arbeit” gehörte. Die häufigen Treffen haben sich dann aber bewährt, insbesondere, weil wir sehr gut unterstützt wurden durch unsere Betreuer, während wir aber gleichzeitig eine grosse Freiheit in der Ausrichtung und Ausgestaltung unseres Lab hatten. Gerade da dieses Lab meine erstere grössere selbständige Arbeit an der ETH war, würde ich die gute Unterstützung und die ideelle Mitarbeit unserer Betreuer nicht missen wollen.

Durch den späten Themenentscheid war unser Lab quasi ein “halbes” Lab: Nachdem wir uns für das Related Work Search Tool entschieden hatten, blieben uns noch sechs Wochen für dessen Umsetzung. Das Risiko, dass wir eingegangen sind, indem wir versuchten Zitierungsdaten zu kriegen, und es lange Zeit nicht klar war, ob das gelingen würde oder nicht, hat sich aber gelohnt! Zeit unnötig verschwendet haben wir wohl, als wir auf die Reviews von Michaels Paper gewartet haben als Entscheidungshilfe, wie wir weiterarbeiten sollen.

Als wir dann schlussendlich den Themaentscheid fällten, war die Einlesezeit eigentlich schon vorbei. Vieles des bereits gelesenen hatte aber nicht wirklich etwas mit dem schlussendlich gewählten Thema zu tun (Small

World, Kleinberg, ...), diese Arbeit war also quasi umsonst (ebenso wie Recherchen zu pdf-html conversion, ...). Umgekehrt mussten wir nach dem Themenentscheid wegen der kurzen verbleibenden Zeit relativ speditiv mit der Arbeit beginnen, weshalb für zusätzliche Lektüre nur nebenbei Zeit blieb; so entdeckten wir viele Dinge auf diesem Bereich, wie z.B. das Kernel-Paper ([5]), erst ärgerlich spät, das schlussendliche Tool blieb algorithmisch simpel.

Ich habe das Lab eigentlich als Kontrastprogramm zu meinen theoretischen Vorlesungen belegt und um Programmierpraxis zu bekommen. Da wir uns die Aufgaben aber ohne viel zu diskutieren nach Fähigkeiten und Interessen teilten, war es schlussendlich Dorian derjenige, der programmierte, und ich las Papers. So hat das Lab für mich einerseits sein ursprüngliches Ziel nicht erfüllt. Andererseits war diese Arbeitsaufteilung die einzig praktikable, vorallem angesichts dessen, das wir schlussendlich in relativ kurzer Zeit etwas auf die Beine stellen mussten. Hätte ich programmiert, wären wir nicht halb so weit gekommen. Immerhin konnte ich Erfahrungen sammeln mit Datenbanken. Die Mischung aus theoretischer Recherche und direkter Anwendung war auf jeden Fall eine reizvolle; im Nachhinein würde ich mir aber mehr Zeit wünschen, so dass wir ein anspruchsvolleres Tool oder einen eigenen Algorithmus hätten gestalten können.

Kritisch betrachtet muss ich aber sagen: es wäre vermutlich auch in der kurzen Zeit des “halben Labs” möglich gewesen, unsere Arbeit besser theoretisch zu untermauern. Rückblickend ist es erschreckend, wie sehr man sich von unbekanntem und riesigen Themengebieten abschrecken lässt. Bei den Themen, in die wir uns zu Beginn einlasen, hatten wir einerseits Lektüreempfehlungen durch unsere Betreuer; andererseits hatte ich in den Bereichen Social Networks und Information Retrieval bereits Vorwissen aus Vorlesungen. So fiel der Einstieg in die Lektüre nicht schwer. Als wir uns in der zweiten Hälfte des Labs aber vom ursprünglichen Themenbereich bewegten und selbst für die Lektürebeschaffung zuständig waren, stand ich im Bereich Co-Citations und Testing/Evaluation plötzlich vor riesigen Themenbergen, bei denen ich nicht wusste, wo mit lesen zu beginnen, während kaum mehr Zeit zum recherchieren da war. Wenn man dann gleichzeitig auch technischeres auf der Todo-Liste hat, stürzt man sich umso lieber da rein.

Sich von grossen, unbekanntem “Themenbergen” weniger abschrecken zu lassen: das war für mich eine wichtige arbeitstechnische Lektion des Labs.

5.2. Erfahrungsbericht Dorian

Ich kann mich Andrea in den wesentlichen Punkten nur anschliessen: Zu viel Zeit ging am Anfang des Labs verloren, während der wir nur relativ planlos Papers lasen, die – wenngleich manchmal äusserst interessant, wie der Artikel von Barabási ([2]) – schliesslich nur am Rande mit unserer Arbeit zu tun hatten. Zu dieser Thematik möchte ich auch noch anmerken, dass ich etwas über die Tatsache erstaunt bin, dass wir von ACM trotz mehrfacher Anfrage bis heute keine Antwort bekommen haben. Man sollte meinen, eine Organisation von dieser Grösse, die für sich in Anspruch nimmt, das Gebiet der Informatik umfassend zu repräsentieren, dürfte in der Lage sein, ihre Mails zu beantworten. . .

Auf der anderen Seite war die Arbeit, sobald die Implementation begann, durchaus spannend. Besonders das Einarbeiten in die verschiedenen Frameworks, die ich vorher alle noch nie benutzt hatte, war erstaunlich interessant und abwechslungsreich. Dazu kann man auch wieder einmal konstatieren, dass gute Dokumentation das A und O für die Produktivität beim Arbeiten mit einer Library ist. Während die ganze Datenbankankündigung mittels des JDBC driver von MySQL schnell und problemlos funktionierte und für *tagsoup* immerhin viele brauchbare Beispiele verfügbar waren, gestaltete sich das Kennenlernen von *JUNG* als etwas schwieriger. Vorbildlicherweise ist zwar eine ziemlich vollständige Javadoc-Dokumentierung vorhanden, die sich jedoch teilweise als sehr knapp erwiesen hat; aber ein Tutorial oder kleine, einsteigerfreundliche Tutorials werden schmerzlich vermisst. Hinzu kamen einige schwer reproduzierbare Fehler, die letztendlich auf Timing-Finessen innerhalb der Visualisierung zurückzuführen waren (“muss `layout.initialize()` jetzt vor oder nach `visuals.setGraphLayout(layout)` aufgerufen werden..?”). Ein angenehmer Nebeneffekt dieses Labs lag in der Auffrischung meiner SQL-Kenntnisse und der Arbeit mit Datenbanken.

Zur Zusammenarbeit mit Andrea kann ich nur Gutes sagen, ich denke, wir haben uns in diesem Projekt gut ergänzt. Wo Andrea sich mit dem theoretischen Hintergrund der von uns verwendeten Konzepten beschäftigte, lag mir eher die praktischere Arbeit der Implementation. Natürlich kamen wir mit dem Näherrücken des Abgabetermins durch die vertane Zeit in eine erhebliche Druckphase, von der wir aber beide relativ unbeeindruckt blieben und die verbleibende Arbeit in einigen Nachtschichten noch erledigten. Mir bleibt am Ende allerdings auch der Eindruck, dass wir mit etwas besserer

Planung und einem früheren Start ein auch algorithmisch anspruchsvolleres Resultat hätten erreichen können.

Die wöchentlichen Treffen mit den Betreuern habe ich sehr geschätzt, da man dadurch etwas weniger Gefahr läuft, seine Arbeit zu vernachlässigen. An dieser Stelle möchte ich mich auch gleich noch bei Yvonne, Michael und Olga bedanken, die Betreuung war ausgezeichnet. Für meinen Geschmack hätten sie ruhig noch etwas fordernder auftreten können. Mir ist bewusst, dass der Sinn des Labs auch im selbständigen Erarbeiten einer Zielsetzung besteht, dennoch wäre mir etwas mehr Führung entgegengekommen.

6. Schlusswort

Eine typisches Schlusskapitel würde Testergebnisse und Evaluationen diskutieren und daraus Schlüsse ziehen. Klassisches Testen war für uns aber aus zwei Gründen nicht möglich: Zum einen macht unser Tool durch das Aufzeigen einer Struktur anstelle eines Rankings keine präzise Aussage, Qualitätsmasse aus dem Information Retrieval wie Recall und Precision lassen sich also nicht anwenden. Zum anderen stellte uns eine “intuitivere” Evaluation (“hilft einem das, was das Tool anzeigt, weiter? Stechen einem die richtigen Papers ins Auge?”) wegen unseres partiellen Datensets vor grosse Probleme: Sowohl ein “Expertenset”, also eine Liste, welche Papers für eine gewisse Query ungefähr wie prominent im Graph erscheinen müssten, wie auch eine Vergleichsliste von Quellen wie Google Scholar oder Libra hätten wir erst via Datenbank auf ACM-Publikationen reduzieren müssen. Vergleichslisten mit anderen Quellen hielten wir ausserdem für zweifelhaft, da die Kritik an deren Rankings einen Ausgangspunkt für unsere Arbeit bildete. Als sinnvollste Evaluationsmethode erschienen uns Expertenlisten, die man thematisch auf ACM-lastige Bereiche hätte beschränken können, wie z.B. auf Computergrafik-/SIGGRAPH-Publikationen. Um solche Listen zusammenzustellen fehlte uns dann schlicht die Zeit (und das Expertenwissen).

Da wir unser Tool von Anfang an eher als “Proof of Concept” denn als fertiges Arbeitswerkzeug ausgelegt haben, ist die Evaluation aber auch, was nun (vielleicht) kommt, indem Personen das Tool anschauen, die mehr von den dargestellten Themen verstehen als wir: Ist die Information, die wir via Graph darzustellen versuchen, verständlich? Bringt die Graphform mehr als ein Ranking, oder, im Gegenteil weniger? Sind Co-Citation und Bibliographic Couples so zentrale Ähnlichkeitsmasse wie wir sie dargestellt haben, oder haben wir sie überbewertet? Lohnt es sich, unseren Ansatz weiterzuverfolgen oder nicht?

Unsere eigene Einschätzung der letzten Frage ist ambivalent: Einerseits scheint die Graphendarstellung viel Information intuitiv zu vermitteln. Es ist aber auch vorstellbar, dass ein Ranking schlussendlich einfacher zu lesen

und informativer ist. Eine grosse Schwäche übernimmt unser Tool von Co-Citations und Bibliographic Coupling: Wo keine solche Couples vorhanden sind, lässt sich keine Aussage über Ähnlichkeit von Publikationen machen. Hier scheint der Ansatz in [5] vielversprechender.

A. Appendix

A.1. Datenbank

Dieser Abschnitt enthält einige Informationen zu nicht selbsterklärenden Tabellen der Datenbank zuhanden von Personen, die mit dieser Datenbank weiterarbeiten wollen.

A.1.1. tblbc

Diese Tabelle führt für jedes Paar von Publikationen den Wert für das “bibliographic coupling” auf, d.h., wieviele Publikationen von beiden zitiert werden, falls dieser Wert grösser als Null ist. Dabei ist jedes Tupel nur einmal aufgeführt; sucht man also nach einem einzelnen Paper, müssen beide Spalten, `a_id` und `b_id` durchsucht werden. Doppelpublikationen mit identischem Titel wurden herausgefiltert.

Die Tabelle wurde mit folgenden SQL-Statements erzeugt:

```
CREATE TABLE tblbc (  
  a_id bigint(20),  
  b_id bigint(20),  
  bc int,  
  CONSTRAINT tab_bc_pk PRIMARY KEY(a_id, b_id)  
)  
  
INSERT INTO tblbc(a_id, b_id, bc)  
SELECT a.fromID AS a_id, b.fromID AS b_id, COUNT( a.fromID ) AS bc  
FROM tblcitationedges AS a, tblcitationedges AS b, tblpub AS c,  
tblpub AS d  
WHERE a.toID = b.toID  
AND a.fromID < b.fromID  
AND c.ID = a.fromID
```

```

AND d.ID = b.fromID
AND c.title NOT LIKE d.title
GROUP BY a.fromID, b.fromID

CREATE INDEX idx_second_ID ON tblbc(b_id)

```

A.1.2. tblccc

Diese Tabelle führt für jedes Paar von Publikationen die Anzahl Co-Zitierungen auf, d.h., wie oft es gemeinsam in einer dritten Publikation zitiert wird, falls dieser Wert grösser als Null ist. Wie bei der vorausgehenden Tabelle ist jedes Tupel nur einmal aufgeführt; sucht man also nach einem einzelnen Paper, müssen beide Spalten, `a_id` und `b_id` durchsucht werden. Doppelpublikationen mit identischem Titel wurden herausgefiltert.

Die Tabelle wurde mit folgenden SQL-Statements erzeugt:

```

CREATE TABLE tblccc (
a_id bigint(20),
b_id bigint(20) ,
ccc int,
CONSTRAINT tab_pk PRIMARY KEY(a_id, b_id)
)

INSERT INTO tblccc(a_id, b_id, ccc)
SELECT a.toID AS a_id, b.toID AS b_id, COUNT( a.toID ) AS ccc
FROM tblcitationedges AS a, tblcitationedges AS b, tblpub AS c,
tblpub AS d
WHERE a.fromID = b.fromID
AND a.toID < b.toID
AND c.ID = a.toID
AND d.ID = b.toID
AND c.title NOT LIKE d.title
GROUP BY a.toID, b.toID

CREATE INDEX idx_second_ID ON tblccc(b_id)

```

A.1.3. tblca

Diese Tabelle führt für jedes Paar von Publikationen, das gemeinsame Autoren hat, an, wieviele gemeinsame Autoren es hat. Wie bei den beiden vorausgehenden Tabellen ist jedes Tupel nur einmal aufgeführt; sucht man also nach einem einzelnen Paper, müssen beide Spalten, `a_id` und `b_id` durchsucht werden.

Die Tabelle wurde mit folgenden SQL-Statements erzeugt:

```
CREATE TABLE tblca (  
  a_id bigint(20),  
  b_id bigint(20) ,  
  ca int,  
  CONSTRAINT tab_pk PRIMARY KEY(a_id, b_id)  
)  
  
INSERT INTO tblca(a_id, b_id, ca)  
SELECT a.id_pub AS a_id, b.id_pub AS b_id, COUNT(a.id_pub) AS ca  
FROM tblpubauthors AS a, tblpubauthors AS b  
WHERE a.id_author = b.id_author  
AND a.id_pub < b.id_pub  
GROUP BY a.id_pub, b.id_pub  
  
CREATE INDEX idx_second_ID ON tblca(b_id)
```

A.2. Einige exemplarische Datentabellen

A.2.1. Die Konferenzen mit den meisten Zitierungen pro erfasstes Paper

Conference	type	#papers	citations	cit._per_paper
IPSJ Meeting	1	1	20	20.0000
Annual Conference on Computer Graphics (SIGGRAPH)	1	1211	13421	11.0826

ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)	1	376	3189	8.4814
ACM SIGCOMM Conference	1	629	5194	8.2576
Internet Measurement Conference / Workshop	1	30	215	7.1667
AFIPS Spring Conference	1	7	50	7.1429
Conference on Computer Supported Cooperative Work (CSCW)	1	525	3744	7.1314
ACM SIGMOD Conference	1	2199	12121	5.5121
ACM Workshop on Role-Based Access Control	1	82	442	5.3902
Symposium on Principles of Database Systems (PODS)	1	835	4498	5.3868
Symposium on User Interface Software and Technology (UIST)	1	483	2245	4.6480
Symposium on Interactive 3D Graphics (SI3D)	1	237	1054	4.4473
ACM SIGPLAN Conferences and Workshops	1	149	645	4.3289
Aspect-Oriented Software Development	1	117	432	3.6923
Internet Measurement Conference	1	148	498	3.3649
Mobile Ad Hoc Networking and Computing	1	218	733	3.3624
Operating Systems Design and Implementation (OSDI)	1	146	482	3.3014
ACM SenSys	1	175	555	3.1714
International Symposium on Physical Design (ISPD)	1	350	1073	3.0657
Design Automation Conference (DAC)	1	3416	10379	3.0383
Domain-Specific Languages (DSL)	1	13	38	2.9231

SIGPLAN Conference on Programming Language Design and Implementation (PLDI)	1	649	1846	2.8444
USENIX Symposium on Internet Technologies and Systems (USITS)	1	86	240	2.7907
International Conference on Computer Aided Design (ICCAD)	1	1898	5090	2.6818
International Symposium on Low Power Electronics and Design	1	836	2167	2.5921
Computer Human Interaction (CHI)	1	3645	9334	2.5608
Foundations of Software Engineering (SIGSOFT FSE)	1	334	843	2.5240
Non-Photorealistic Animation and Rendering	1	73	184	2.5205
Workshop on I/O in Parallel and Distributed Systems (IO-PADS)	1	30	75	2.5000

A.2.2. Co-Citation Coupling Werteverteilung

ccc	# pairs
1	575322
2	69760
3	20240
4	8415
5	4355
6	2468
7	1565
8	986
9	699
10	450

11	362
12	247
13	211
14	152
15	112
16	89
17	75
18	73
19	42
20	40
21	45
22	32
23	24
24	17
25	15
26	22
27	15
28	12
29	15
30	5
31	7
32	6
33	9
34	1
35	9
36	3
37	4
38	2
39	1
40	2
41	1
42	3
43	3
44	1
45	1
46	1
49	1

50	1
51	1
53	1
55	1
56	1
80	1
130	1
163	1

A.2.3. Bibliographic Coupling Werteverteilung

bc	# pairs
1	1258461
2	184416
3	47838
4	16541
5	6310
6	2678
7	1222
8	655
9	345
10	178
11	87
12	33
13	28
14	18
15	8
16	12
17	3
19	4
20	3
21	2
23	1
24	1
27	1

Literaturverzeichnis

- [1] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 44-54, 2006.
- [2] A.L. Barabási, H. Jeon, Z. Neda, E. Ravasz, A.Schubert, T.Vicsek. Evolution of the social network of scientific collaborations. In *Physica A*,311(3-4): 590-614, 2002.
- [3] Ergin Elmacioglu, Dongwon Lee. On Six Degrees Of Separation in DBLP-DB and More. In *ACM SIGMOD Record*, 34(2):33-40, 2005.
- [4] Birger Hjørland. Core Concepts In Library And Information Science. <http://www.db.dk/bh/Core%20Concepts%20in%20LIS/home.htm>, 2005.
- [5] Takahiko Ito, Masashi Shimbo, Taku Kudo, Yuji Matsumoto. Application Of Kernels To Link Analysis. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 586-592, ,2005.
- [6] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In: *Journal of the ACM*, 36(5):604-632, 1999.
- [7] Michael Kuhn, Roger Wattenhofer. The Layered Small World of Scientific Conferences. Submitted.
- [8] Loet Leydesdorff, Liwen Vaughan. Co-occurrence matrices and their applications in information science: Extending ACA to the Web environment. In *Journal of the American Society for Information Science and Technology*, 57(12):1616-1628, 2006.
- [9] John Martyn. Bibliographic coupling. *Journal of Documentation*, 20(4): 236, 1964.

- [10] Friedemann Mattern. Zur Evaluation der Informatik mittels bibliometrischer Analyse. In *Informatik Spektrum*, 25, 22-32, 2002.
- [11] Friedemann Mattern. Bibliometric Evaluation Of Computer Science. <http://www.sarit.ch/events/SARIT06/Mattern-Bibliometry-SARIT06.pdf> . 2006.
- [12] Michael H. MacRoberts, Barbara R. MacRoberts. Problems of Citation Analysis: A Critical Review. In *Journal Of The American Society For Information Science*, 40(5):342-349, 1989.
- [13] Alireza Noruzi. Google Scholar: The New Generation Of Citation Indexes. In *Libri*, 55:170-180, 2005.