

Releasing the Power of Digital Metadata: Examining Large Networks of Co-Related Publications

David Tarrant, Dr Les Carr, and Dr Terry Payne

School of Electronics and Computer Science, University of Southampton,
Southampton, UK,
`dct05,lac,trp@ecs.soton.ac.uk`

Abstract. Bibliographic metadata plays a key role in scientific literature, not only to summarise and establish the facts of the publication record, but also to track citations between publications and hence to establish the impact of individual articles within the literature. Commercial secondary publishers have typically taken on the role of rekeying, mining and analysing this huge corpus of linked data, but as the primary literature has moved to the world of the digital repository, this task is now undertaken by new services such as Citeseer, Citebase or Google Scholar. As institutional and subject-based repositories proliferate and Open Access mandates increase, more of the literature will become openly available in well managed data islands containing a much greater amount of detailed bibliometric metadata in formats such as RDF. Through the use of efficient extraction and inference techniques, complex relations between data items can be established. In this paper we explain the importance of the co-relation in enabling new techniques to rate the impact of a paper or author within a large corpus of publications.

1 Introduction

Bibliometric techniques have emerged as an important mechanism to identify the significance of articles from the literature, and by extension, the quality of the work described. In the first instance, this is achieved simply by counting the number of citations that a publication receives. Thus the ‘citation count’ becomes one example of a metric which allows a reader to determine a publication’s contribution to a research discipline.

Commercial secondary publishers have typically taken on the role of rekeying, mining and analysing this huge corpus of linked data, but as the primary literature has moved from print to the more open world of the web (in researcher’s web sites or in institutional and subject-based repositories) this task can now be undertaken by new services. Citeseer¹, working on the Computer Science literature found on web sites, Citebase², working on publications in Open Access

¹ <http://citeseer.ist.psu.edu/>

² <http://www.citebase.org>

repositories, and Google Scholar³, drawing from published journal collections as well as the open web, all provide some kind of alternative to commercial citation analyses. As Open Access mandates increase the use of digital repositories, more literature will become openly available through a network of well-managed data islands, exporting their holdings in a variety of metadata schemas and formats such as XML and RDF. With the proliferation of online repositories, the amount of publications available is now far greater than via paper publication. With such a large corpus of data, accurate ranking mechanisms become an important tool enabling key publications within a field to be located.

In recent times the value of metadata has become more apparent with the push for Web2.0 content and sites. Web sites such as flickr⁴ and facebook⁵ allow users to create a profile and then link their profiles to others via arbitrary links. These links can be analysed to define characteristics which can connect many “spaces” or be used as recommender systems to new users. Simple examples include the Amazon website⁶ which suggest additional products bought by customers who purchased the current subject item. In publication networks citation links are guaranteed to exist between articles and these can be further analysed to identify links such as those between authors. Another important mechanism by which citations can be analysed is through the recognition of Co-Relations. Co-Relations represent links between two items which are established by a third, e.g. two papers being cited together (co-citation). Co-Relations grow stronger as the same two items are Co-Related with greater frequency. Co-Relations can establish many facts about a publications local subject tree, such as those papers which share similar characteristics, are in the same research area or are critical background material to a particular subject. These techniques can be useful when classifying newer publications.

The majority of digital repositories which export metadata currently enable access to the core details pertaining to a publication (name, authors, keywords), and the citations which that publication lists. From this data all of the co-citations can be inferred and stored for use, although possible this process becomes challenging when processing several sets of metadata sourced from different locations. With many different mechanisms and systems available for data exportation we look at methodologies provided by the semantic web community for the purposes of data exportation and later alignment.

From the many uses of Co-Relation, this paper looks at extending the theory behind finding similar publications in a subject area to being able to use this to rank the impact of the publication. We propose a new mechanism “CoRank”, which identifies significant papers at an early stage in the publication life cycle. We present the algorithm and discuss the implementation showing empirically that this achieves significant gain over existing techniques to identify high impact papers within a field. In tests it is shown that CoRank achieves an accurate

³ <http://scholar.google.com>

⁴ <http://www.flickr.com>

⁵ <http://www.facebook.com>

⁶ <http://www.amazon.com>

ranking for a paper within 12-18 months rather than the more typical two to three years, a 50% improvement. We also present Co-Pilot, a system which makes use of pattern templates acting upon semantic data to identify features in open archive resources that can be used with bibliometric methods. More specifically Co-Pilot has been provided with a Co-Relation template which not only enables us to infer co-citations but also further Co-Relations such as Co-Authorship and Co-Institutional relations.

The paper is organised as follows: In section 2 we give a background of bibliometric analysis and discuss various current approaches for generating ranking data for a given set of papers. In Section 3 we present our method for inferring co-relations and demonstrate how pattern templates have many advantages over querying and reasoning techniques. Section 4 outlines the CoRank algorithm and discusses our implementation. Section 5 presents an initial empirical analysis of our system and we conclude in section 6.

2 Bibliometrics

Bibliometrics began as the statistical study of bibliographies and was initially termed ‘statistical bibliography’ before the term ‘bibliometrics’ was proposed in 1969 by Pritchard (1). By looking at the bibliographies of scholarly literature, a vast network of academic papers can be constructed through the links created by citations and footnotes. Bibliometrics has now become an abstract term for data processing techniques involving more than just the bibliography of a document. Alternative terms are starting to appear including ‘scientometrics’ and ‘informetrics’ (2), which all cover the study of large network graphs constructed from links between (often scholarly) material (3) (4). The purpose of such graphs is to enable production of quantitative estimates of the importance and “impact” of individual scientific papers, journals and authors. In Bibliometrics, the best known technique — currently the most widely used and respected — is Garfield’s impact factor (5). This is used to provide a numerical assessment of journals and is a measurement of Journal Impact Factor (JIF). This idea developed into what we know today as the Science Citation Index (SCI)⁷, which currently stores citation data relating to over 5,800 journals. More recently JIF has been joined by Article Impact Factor (AIF) and Position Impact Factor (PIF) which attempt to analyse the standing of individual pieces of scholarly literature. Typically an AIF is calculated from the number of citations which the article in question receives. It has been shown however that a raw count is not the best approach to use when trying to distill large amounts of information, especially on the web. With early mechanisms only looking at citation count, authors on the web found they could create a high ranking web site by simply creating false linking pages which bear no context to the actual article.

To try to circumvent this problem work has been done to find the authoritative sources of information (6) (7), perhaps the best known of which is PageRank

⁷ <http://scientific.thomson.com/products/sci/>

(8). PageRank provides one of the best impartial algorithms and uses statistical probability to take into account how a typical user would “surf” the web. Currently PageRank provides the basis of the Google searching and indexing system (9). On the web, bibliometric analysis is limited to the domain of citations. With scholarly articles, we are presented with a much richer source of ‘correct’ information which allows establishment of relationships between authors, topics and institutions as well as between the articles themselves.

Other early problems realised by Garfield and the SCI include ambiguity, where names of items can be abbreviated and thus misunderstood. This is just one example of a problem which the Semantic Web attempts to solve by providing all objects a unique ID, a concept which we rely on in the scope of this research.

Article Impact Factor ranks each article individually. By taking some of the ideas from the SCI and web paradigms such as PageRank the AIF has become a respected way of ranking research. AIF techniques are also very important when it comes to publication in online repositories in addition to or instead of Journals. Open access movements have recently lead to enhancements in free availability of research in online repositories, averting the need for libraries and academic bodies to subscribe to an increasing number of journals. JIF can provide an early indication of a publications later impact; a paper published in a high impact journal is likely to be a high impact paper, the same cannot be applied to electronic repositories. These repositories are not as heavily peer reviewed as journal publications, and each repository can contain articles covering many subject areas.

In recent times, measurements of AIF have become increasingly important in web search engines and many of the same techniques can be applied in both areas of ranking academic publications and web based documents. Citation analysis works on the assumption that influential scientists and important works will be cited more than others. More recent studies on this data have shown that that a simple citation count technique is able to effectively rate the impact of papers in the same way as the same set of papers being peer reviewed (10) (11). Many sites now exist to enable easy location of high AIF publications including: Google Scholar (12) (based on PageRank), Scopus (13) and CiteSeer (14). To enable bibliometric analysis a citation network has to be constructed. In this paper we are going a step beyond the citation network and creating the Co-Citation and Co-Relation networks.

3 Citation Networks

The Citation network of a particular paper grows over time as a publication is cited directly by other publications, (Figure 1). The set of papers (A from equation 1) grows gradually over time and each individual paper provides a single piece of extra metadata by which the target publication can be ranked.

In the co-citation network (Figure 2), each additional publication which directly cites our target publication also provides metadata relating to all other

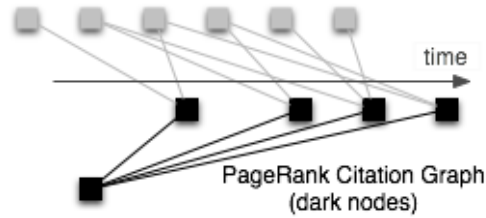


Fig. 1. Citation Network

publications which it cites. These publications represent the set of publications (P_x from equation 2) with which we are co-cited.

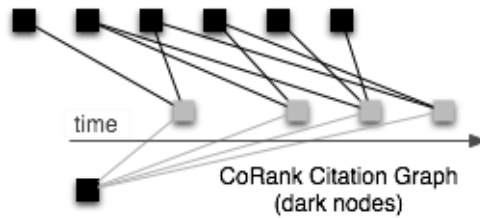


Fig. 2. co-citation Network

Citation graphs build as a publication is cited by others. Although a citation graph will continue to grow, there is a period of time which dictates the most accurate measurement for the maximum impact rating the article will achieve. This point occurs just as the article achieves its peak citation rate: the most citations in the shortest amount of time. In many subject areas the time in which a publication will reach its peak citation rate is predictable with a certain degree of accuracy. This predictability is all down to how researchers in a subject area operate; factors such as journal publication schedules, rate of open access publication and conference occurrence all affect how fast a field of research moves forward. For the majority of subject areas the time which an article takes to reach its peak citation rate is just under 3 years (15).

Although ePublication is reducing the time taken for an article to reach its peak citation rate (16), there is still the opportunity for research into methods of ranking publications via the use of earlier indication metrics. Brody (16) investigates the use of download metrics to achieve this, concluding that download metrics do afford a good early indicator providing the field of research uses distribution techniques which are well controlled. Download counts are much like journal subscription counts where the distributor knows the number of journals it sends out. If however a publication is available in many disparate locations it is difficult to pool the results from these sources.

In this paper we look at a method of ranking the impact of papers through the usage of enlarged citation graphs, which can be built from extended metadata. By using reasoning techniques which look for patterns within the data we can quickly build an article’s co-citation graph. We start with the set of papers A containing all articles a which cite publication p (Equation 1).

$$A = \{\exists a \wedge (a \text{ cites } p)\} \quad (1)$$

As well as locating the articles A which directly cite the publication p , a co-citation graph P also includes all articles which are cited by the set of articles A (Equation 2).

$$P_x = \{\forall a (a \in A) \wedge (a \text{ cites } x)\} \quad (2)$$

With the set P_a typically containing around 20 articles the co-citation graph of P becomes larger than the citation graph of A in a much shorter time.

Co-citation analysis of the graph (P) is traditionally used to relate two objects together by saying that they are strongly linked in some way. As an example, if two papers are highly co-cited then we can say that these papers are related and/or core material to this subject area (17). Co-citation relations can also be applied effectively in other situations such as social networks where users can build a network of close and not-so-close friends and relate items like favourite pictures and music in their profiles.

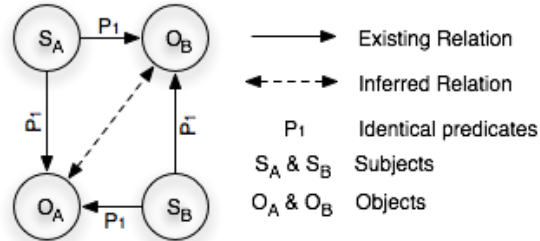
In the scope of this document we look at how the co-relation graph (the most general form of P) can be obtained from raw metadata where these relations are not present (Section 3.1) and then introduce CoRank (Section 4). Section 4 gives an overview of some first generation CoRank algorithms which are being looked at to provide a technique for ranking the impact of papers earlier in the publication life cycle. It is hoped that by looking at relations obtained between the publication in question and other articles we can obtain an approximate placement for where the publication sits within its research hierarchy.

3.1 Inferring Co-Relations

The metadata available from publication records typically consists of out-facing links only: those which state this publication’s relations to others, not other publication’s links to this one. For the purposes of this paper we are calling these out and in-facing links respectively. To find all of the publications with which our target publication is co-cited we first have to find all of the publications which cite it; this involves finding all of the in-facing links. The Co-Pilot system outlined later, operates at a level above this and takes in a pattern which it is able to infer from the dataset. The Co-Relation pattern in its most basic form represents a relation between two items which is established by a third (Figure 3).

Note that the type of the Subjects (S_A and S_B) and Objects (O_A and O_B) must be the same for the co-relation pattern to hold, for example the Subjects could be Papers and the Objects could be People. This combined with the has-author predicate would enforce the co-authorship relation.

Fig. 3. The Co-Pattern in its most general form



To enable the co-relationship to become useful you not only need to know that it exists but how many times it occurs. The relation between items becomes stronger the more times that they are cited together.

By using technologies, such as RDF, provided to us by the Semantic Web, we can start to align large corpuses of data based upon the definitions of objects and relational predicates outlined in an ontology. An ontology provides a detailed computer readable description of the many object and link types which are used in an RDF document. An ontology allows validation of the input document as well as inferences to be performed based upon relationships between objects outlined in the ontology document. A system level ontology can also be employed to outline alignments between many ontologies and namespaces, thus data from both the ACM and IEEE dataset can be used in conjunction with each other.

By using ontologies to describe namespaces and relations between items, reasoners can then be used to parse input documents and output any inferences which can be deduced. Due to the complexity of relations which can be inferred, a reasoner has to have the entire input document in working memory in order to successfully compute full deductive closure. Having to keep all of the information in working memory means that each time the input document changes it has to be reparsed in its entirety.

Another method by which co-relations can be located is by writing specific queries to locate the data. In traditional Relational Database Management Systems (RDMS) this would be a time consuming but relatively easy process; the query is limited to using the column definitions of the tables stored within the database. SPARQL (RDF Query Language) provides a similar set of features as that of SQL for RDMS but due to the lack of Unique Name Assumption⁸ is it not possible to gain a grouped count of unique IDs from a SPARQL query. In respect to the Co-Relation pattern, it is not possible to use either SPARQL or reasoner's to locate all Co-Relations without providing many queries or definitions, parsing each one individually.

We have outlined two problems here. Firstly, finding the co-relation via rules or queries is intractable on large and ever-changing data. Secondly, existing sys-

⁸ On the World Wide Web (WWW) unique names assumptions are not valid, because hosts may have more than one name, and files may have multiple links to them.

tems do not perform any type of caching upon the input documents and thus have to reparse the entire input document again if any one piece of information changes. The solution proposed here has thus been created to solve both of these problems whilst not limiting the user to the confines of the designed system.

3.2 The “Co-Pilot” system

Co-citation questions are difficult to answer, and so the Co-Pilot system was designed to help answer the most extreme of those questions: *which papers are co-cited the most?*

The Co-Pilot system is designed to parse input documents sourced from specific namespaces. This enables Co-Pilot to employ stateful caching between versions of the document and only parse records within the document which have changed. Co-Pilot assumes that deductive closure via reasoning has already been performed on the input document prior to it being passed to Co-Pilot. Co-Pilot can thus be “plugged” into existing semantic storage systems such as Jena⁹ or used as a stand alone application feeding back RDF documents containing only the new instances to the user.

As outlined in the previous section all instances of co-citations can be retrieved directly from a triple store using a SPARQL query similar to that shown in Figure 4. This query will return all instances of co-citations and you can filter out where *?x* and *?y* (unique ids for publications) are the same. However, due to the UniqueNameAssumption which exists in SPARQL a quantified grouping of matching co-relations between *?x* and *?y* for differing *?p* cannot be found without further processing.

```
?p ?cites ?x
?p ?cites ?y
?x type ?z
?y type ?z
```

Fig. 4. All co-citations Query

Taking the more general form of the co-relation pattern the next stage is to find all instances of this pattern for all objects (*?p*) which exist within the data. Rather than supplying *cites* as our predicate to the query we now want it to find all instances where the pattern from figure 3 holds true, for values of *?p*, *?x* and *?y* of any type *?z*, and any predicate *?pred* (figure 5).

Co-Pilot has been designed to specifically handle the pattern supplied in figure 5 within a linear time frame and store this in a caching system for later retrieval. Co-Pilot stores the Co-Relation pattern in such a way that the primary index represents an instance of the pattern. Subjects are then related to the first

⁹ <http://jena.sourceforge.net/>


```

?p ?pred ?x
?p ?pred ?y
?x type ?z
?y type ?z

```

Fig. 5. The General Co-Relation pattern

occurrence of the pattern and any additional occurrences simply increment the number of times that particular instance has occurred. This means that each Co-Relation can occur many times but is only indexed once within the database, thus providing an optimal data structure from which co-relations can be retrieved. While parsing, if a single record in a multi-record document has not changed since the last parse Co-Pilot will not require a complete re-parse to update its data. Thus it is ideally suited for use with frequently changing input data.

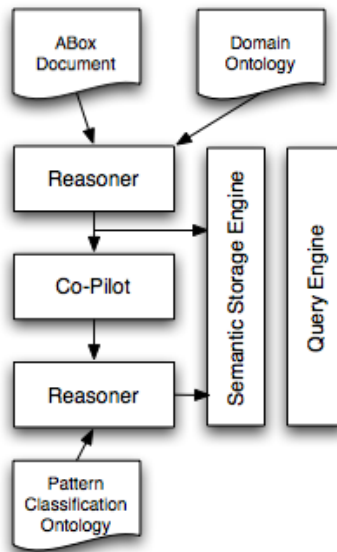


Fig. 6. Co-Pilot in Context

Figure 6 shows where the Co-Pilot system fits in the process diagram of a semantic storage engine and query system. Co-Pilot is designed to take in RDF and export RDF back to the user. It achieves this by providing a series of services which can be invoked to import and export data relating to documents or namespaces. To enable the system to remain stateful between parses the Co-Pilot system maintains its own heavily indexed data structure (cache) which is specific to the co-relation pattern. Co-Pilot performs all the processes which

enable it to only process input data which has changed and manages any required threading allowing the input document to split into smaller chunks which are then processed in parallel. Finally Co-Pilot can be asked to export all of its inferred data back into RDF as a series of Co-Patterns, which can be processed by 3rd party reasoners.

Currently the Co-Pilot system is heavily customised specific to the aim of locating co-relations with a set of input documents; further research could be done to find a method for translating the specific classes defined in an ontology into the most generic patterns. These new patterns could be parsed to Co-Pilot which would need to apply logical methodologies to find the quickest way to parse the dataset, and how to store the inferred instances in an indexed data structure. Although Co-Pilot is limited to the Co-Relation pattern, the caching facilities and increased speed of execution provided demonstrate the capability of such systems to be the basis for newer more powerful applications.

3.3 Co-Pilot Performance

In order to judge the impact of publications effectively, a well maintained network of associations between these publications is needed. In practical terms a citation service, such as citebase, will be harvesting new publications appearing in repositories on a continual basis. For that reason, we need a system which that allows incremental updates to be made without regenerating the entire citation graph. CoRank requires a complex network of associations which not only relates publications to their direct peers, but also to the articles which their peers cite. Co-Pilot was designed to construct this network quickly and efficiently taking on the role of quantifying, managing and storing the data such that it can be both updated and exported. Due to the negligible differences found in testing small datasets, it was decided to test Co-Pilot on a dataset containing just under 4,000,000 RDF triples. This represents a reasonable sized dataset of publication data dating back to the late 1980's. Comparative tests were performed between Co-Pilot and the RDF storage engine 3store (v3.0.17)¹⁰ for both importing and querying data. The major difference between the two systems is that once Co-Pilot has imported the data, all instances of co-relations have to be found, indexed and quantified. 3store, unlike Co-Pilot, provides an interface which can be used to query the imported data using the SPARQL query language, however no relations have been inferred at this stage. Table 1 outlines the time taken to first import the data, followed by the time to output a document containing the co-citations.

The co-citation results obtained from the 3store dataset (marked *), contain all instances of co-citations, however these are not grouped or quantified into unique occurrences of a co-citation. An extra stage would be required to perform this operation adding to the execution time required to obtain the correlated set of results. Co-Pilot outperforms 3store on the import of the initial data however this increase in performance can be considered relative since 3store

¹⁰ <http://www.aktors.org/technologies/3store/>

| System | Import Time | Triples/Second | Output Time |
|----------|-------------|----------------|-------------|
| Co-Pilot | 35 minutes | 1876 | 10 minutes |
| 3store | 60 minutes | 1094 | 6 minutes* |

Table 1. Co-Pilot vs 3store: Import and Retrieval Times

performs basic RDFS entailment reasoning as the data is imported. A semantic storage engine such as 3store is necessary if the original data and that output from Co-Pilot needs to be queried at a later stage. Working on the assumption that a single input document contains all data pertaining to publications from a single source, if one is added or adjusted this document has to be re-imported. Due to the built in caching within Co-Pilot this process is much simpler than than the same process in 3store which looks for entailment which no longer hold as well as new ones which are established by new data. This goes some way to explaining the figures in 2, which represents the comparison in time taken to re-import the same document with an author addition on a single publication.

| System | Import Time |
|----------|-------------|
| Co-Pilot | 1m40s |
| 3store | 1h 34m |

Table 2. Co-Pilot vs 3store: Re-Import Times

To aid in lowering the update time of a semantic storage layer application, such as 3store, Co-Pilot is also able to export only the changes to the original document since the last export. Co-Pilots increased performance in re-importing allows quick and efficient output of an updated graph of co-relations. This provides the input to the CoRank ranking process.

4 New Bibliometrics

Publication ranking and impact is typically measured in simplistic ways based upon the number of citations a publication receives. With more and more publications now finding their way into hubs of information through the use of electronic repositories, effectively ranking documents against each other requires an algorithm more complex than a simple citation count. In this section we look at how Google’s PageRank algorithm (8) provides a good solution by which publications can be ranked in online repositories and how this can be improved by looking at the larger network created from the Co-Relations. In this section we present the CoRank algorithm which takes the PageRank algorithm one step further and attempts to rank publications accurately sooner in the publication lifecycle. By using the larger network built on the Co-Relations we conclude that the rank for a publication stabilises in a much shorter amount of time than that required for PageRank.

PageRank is the underlying algorithm behind the success and impartiality of Google. PageRank works by ranking every page based upon every other page which links to it, this builds a large network of links relating each page to others. This technique means that PageRank is very similar to a plain citation count as used in current bibliometrics; the key difference is that you do not receive a score of one for each citation you receive. The score which a page receives from a citation though PageRank is the rank of the page which cites you divided by the number of other pages it also cites. By using this technique PageRank is able to eliminate false citation information such as that received from a publication which cites hundreds or thousands of other publications; the same applies for self citing from other low ranking pages.

The PageRank algorithm (equation 3) is applied iteratively over a citation network to rank each page. For the rank of a document to reach a stable point the iteration has to be performed at least 5 times, and up to 10 in larger systems. Before the algorithm is first run the rank of each paper is set to $1/|V|$ where $|V|$ is approximately the number of publications/pages in the system. A damping factor can also be used which is designed to represent the probability of a person following a link to continue to your page (represented here as α). From research performed by Brin/Page (9), it is recommended to set this figure to 0.85. With the dampening part of the algorithm fixed, the PageRank of a paper is generated from the PageRank of the papers p_j which cite the original paper divided by the number of papers (L) which p_j cites.

$$PR(n) = \frac{1 - \alpha}{|V|} + \alpha \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)} \quad (3)$$

In a closed system such as a publication repository, the dampening factor and initial division by the number of publications in the system also has the positive side effect that if a publication is not linked to by anything its rank will not be set to zero (providing that the system catches the division by zero). Although PageRank is a very effective method of ranking articles in a particular search, it takes a while for an article to gain enough citations for the rank to stabilise. This will be a little while after the paper has reached its peak citation rate which can take beyond 2 years.

The CoRank algorithm is a logical step beyond PageRank and utilises larger network graphs (Figure 2) constructed from the co-relations. Using this larger network we can begin to research techniques which are able to analyse and potentially rank a publication much earlier in its life cycle. Equation 2 outlines the core CoRank algorithm. Again, the dampening factor is included to avoid a publication's rank being set to zero.

$$CR(p) = \frac{1 - \alpha}{|V|} + \alpha \sum_{cp_j \in M(cp_i)} \frac{CR(cp_j)}{CL(cp_j)} \quad (4)$$

Like PageRank, the initial CoRank value is set to $1/|V|$. The CoRank of a paper (p) is generated by taking the CoRank ($CR(cp_j)$) of each paper p is co-cited

with and dividing this by the number of papers ($CL(cp_j)$) this paper (cp_j) is co-cited with. An iterative parse is then performed over all of the objects in the dataset to find their new ranking figure. correlated At the end of each parse the figures are scaled such that the highest ranked paper has a rank of 1.

5 Results

For the purposes of the evaluating the effectiveness of CoRank, the CiteBase ¹¹ dataset has been used as our exemplar dataset. Citebase performs many of the ranking tasks (such as PageRank) already and thus has all of the direct citation links between papers already indexed. Citebase holds articles from physics, maths, information science, and biomedical science and contains over 200,000 publications. A snapshot of this dataset was taken in March 2007 containing 263,619 publications and from this 36 previous monthly snapshots were generated with the first one (March 2004) containing 174,786 publications. PageRank and CoRank were then run upon each dataset for six iterations to produce the ranking values for each publication in the dataset. Finally nine samples of 100 papers have been taken which first appeared in the dataset in the first few snapshots; representing a selection of three year old papers. The ranking data from these has been used in this section to track the average rank growth and correlation to PageRank and cite count rank order, with the top 500 being taken for the purposes of rank order comparison. In this section we are looking for CoRank to provide an early indication of impact after the PageRank and citation count life cycles have completed. We are also looking for the average age of the top 500 papers by CoRank to be younger than that given by PageRank and citation count.

With the indexes constructed using Co-Pilot, CoRanking the snapshot takes negligibly more time than PageRank ranging from 14-26 minutes dependent on the size of the initial dataset. Configuration of the machine running the Ranking is a 3.2GHz Xeon with 4GB of RAM which is able to load the entire dataset into memory for the purposes of ranking.

5.1 Ranking Algorithms

In this section we present comparisons between existing algorithms against early implementations of CoRank. We look at how quickly a paper's rank, or Article Impact Factor (AIF), stabilises with respect to each algorithm and compare this to the rank order of other algorithms. We are working on the assumption that three years is a suitable time period over which to take these readings, as the majority of publications should reach their peak citation rate within this time. Figure 7 plots the average rank of the three metrics against each other over the three year period. While citation count remains fairly linear on average, PageRank and CoRank fluctuate as the graphs from which their data is obtained

¹¹ <http://www.citebase.org>

stabilise. CoRank shows a much clearer stabilisation leading up to the end of the three year time frame, this reduction in gradient occurs between the 12 and 18 month old period in the publication life cycle.

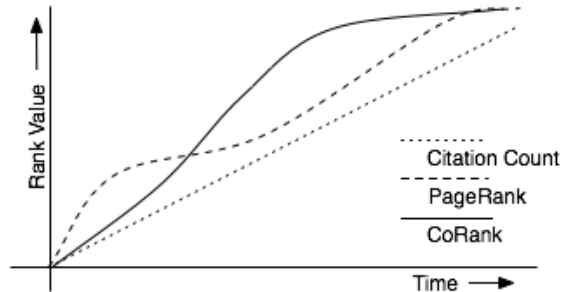


Fig. 7. Citation Count, PageRank & CoRank average (over three years)

Working on the hypothesis that Citation Count and PageRank are effective means by which publications can be ranked, we are going to compare the rank order, or Position Impact Factor (PIF), of each algorithm. From our assumption that three years is enough for the rank order to stabilise, we are going to take the rank order from the third year snapshot as our target. Running each algorithm over the previous snapshots and extracting the order of the same 500 publications allows us to analyse if CoRank is correlated to the final ordering at any time earlier in the publication life cycle. In order to compare the rank order of the two datasets we employ Spearman’s rank correlation coefficient (Equation 4) which takes the difference (d) between the ranks of 2 items and calculates the correlation based upon the number of items (n) which exist within the system.

$$p = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (5)$$

The Spearman rank (p) for the dataset which will range from -1 to 1 with -1 being a perfect negative correlation and 1 being a perfect correlation; 0 represents no correlation at all. For the purposes of analysing the results here we are going to translate these figures into a percentage figure. Table 3 demonstrates the maximum stabilised match of each algorithm to the target algorithm (shown in column 1). The readings show the percentage match and the age (in months) at which this was achieved.

| Target | Cite Count | PageRank | CoRank |
|------------|------------|-----------|-----------|
| Cite Count | | 40% @ 24m | 10% @ 5m |
| PageRank | 25% @ 24m | | 10% @ 20m |

Table 3. Spearman Rank Comparison 1

The surprising results from this table (Table 3) show that although Citation Count and PageRank are well respected and established algorithms they don't actually perform that well against each other. CoRank performs badly against both PageRank and citation count here which although surprising can be explained by the average age of the papers in the top 500.

5.2 Improving CoRank

The current CoRank algorithm takes into account all papers with which a publication is co-cited, no matter how weak the link to that publication is, this is meaning that the older papers with a greater frequency of strong linked are ranked higher than newer papers. In CoRank 83% of papers are older than 2 years which shows very little improvement on the 95%+ of citation count and PageRank. Weak citation links can be ruled out by taking a top percentage of papers a publication is co-cited with. This led to an improvement in average paper age, with only 41% now being older than 2 years, however the top 500 correlation was still weighted heavily towards a certain set of papers in our dataset. This collection of about 100 papers in the dataset were gaining very high ranks and which affected the overall rankings within the result set when scaling. This set of review papers gained a constant high ranking in each snapshot due to the gap in co-citation count between themselves and every other paper in the system. Looking at the pattern of citations gained by the review papers it was found that they had passed their peak citation rate but this wasn't being taken into account by the CoRank algorithm. Equation 6 shows a simple iterative extension to CoRank which takes account of the age of the co-citation when calculating the rank.

$$CR(p) = \frac{1-\alpha}{|V|} + \alpha \sum_{cp_j \in M(cp_i)} \left(\frac{CR(cp_j)}{CL(cp_j)} / X_{age+1} \right) \quad (6)$$

In this equation, X is able to either represent the paper cp_j , with which the target publication is being co-cited, or the paper which establishes the co-citation between an arbitrary publication and the target. Thus one represents the age of the co-cited paper and the other represents the age of the co-citation, here named $CoRankTime(cp)$ and $CoRankTime(p)$ respectively.

| Target | $CoRankTime(cp)$ | $CoRankTime(p)$ |
|------------|------------------|-----------------|
| Cite Count | 35% @ 3m | 40% @ 8m |
| PageRank | 20% @ 20m | 5% @ 8m |

Table 4. Spearman Rank Comparison 2

Table 4 demonstrates both algorithms performing well against citation count, stabilising in a very fast time to a PIF index correlated positively to Citation Count. Though neither performs particularly well when compared to PageRank,

a strong positive correlation to Citation Count is established very quickly. This demonstrates that potential high impact papers after three years can be found as early as three months into the publication life cycle. Being based upon citation and co-citation age allows the older but still frequently cited papers to still be highly ranked. Both algorithms showed a very similar distribution in paper age with 40% being older than two years and a 60:40 distribution on the remaining between 1-2 years of age and less than 1 respectively.

6 Concluding Comments

Bibliometrics provide the basis for analysing research for classification, categorisation and impact ranking. Existing techniques such as citation count and PageRank provide techniques by which impact of a publication can be measured and both of these examples are in widespread use today. Citation count provides a simplistic mechanism which rates a publication higher dependent purely on the number of other publications which cite it. This is fine until we start to realise that some citations may well come from publications which have purely been written to boost the ranking of the one in question. PageRank goes some way to solving this problem by ranking a publication based upon the rank of each one by which it is cited. Both of these techniques suffer from the fact that a publication takes time to gain an accurate measurement of impact, both of which typically occur around the point when the publication reaches its peak citation rate. Citation count and PageRank are also dependent purely on a network of in-direction links to the publication in question. This graph also takes time to obtain enough data to accurately determine a rank value. CoRank extends this paradigm by operating over a network graph based upon the publications an article is co-cited with rather than directly cited by. This link network grows much larger than that required for PageRank or citation count and this is also achieved in a much reduced time period. A co-citation graph has commonly been used to find which research area or areas a publication is related to, CoRank takes this one step further by looking at where an article is ranked based upon what it is CoRanked with.

Identifying and quantifying co-citations which exist in a given dataset is the job of Co-Pilot. Designed to parse RDF, Co-Pilot is able to parse input documents containing publication metadata locating co-relations within the data structure and caching these for later output. In quantifying the number of times each instance of a co-relation occurs Co-Pilot is able to offer great improvements over existing data querying techniques designed to operate over RDF. By looking for the most general occurrence of a co-relation Co-Pilot is able to identify relations which exist between papers, authors, publishers and any other forms of co-relation which may exist within a system, such as that built by a social network. Co-Pilot provides a series of interfaces able input/update and export data in RDF and once a set of data has been exported an ontology can be used to classify specific types of co-relation patterns. Providing the ontology for the co-citation relation to a reasoner enables a network graph of co-relations relating

to a single publication to be built much quicker than if Co-Pilot was not used in the intermediate stage.

Running CoRank over this network graph results in an 12-18 month improvement on PageRank in terms of time taken from a document ranking to stabilise; a 50% improvement. Although 50% represents a significant improvement it bears no relation to the position rank order comparison between CoRank and citation count/PageRank. Upon further analysis of the dataset and the results obtained it was found that CoRank is very sensitive to change. For papers which continually receive a few citations every now and again, CoRank is able to handle these effectively. Both PageRank and CoRank suffer from the highest ranking papers being rarely cited, but being cited by high ranking publications; this leads to the dataset being weighted heavily towards what appear to be low ranking papers. Introducing an age factor into the CoRank algorithm eliminated these problems and in turn led to a much improved position index factor when compared to both citation count and PageRank. A 35-40% correlation at 3-8 months compared to the target reading taken at 36 months gives an 80% improvement in impact ranking. Introducing the factor which weights the rank based upon the age of citation we also see an even distribution in paper age, where older but still highly cited papers are maintaining a high rank. Such a metric should prove very useful in the user testing phase which is currently in the early planning stage. If CoRank proves itself through user testing and becomes widely used we may start to see publication life cycles becoming shorter as high impact research is revealed much quicker. CoRank is able to extend beyond the area of publications however and can also be looked at to provide metrics on social networks and also in recommender systems.

There is still much ongoing research in the area of advanced bibliometrics and as digital repositories and web2.0 gain more followers we are starting to realise the power of the available data. By borrowing technologies from these areas as well as the Semantic Web we can start to realise some of these possibilities and their possible future impacts. The Co-Relation is a very important pattern in modern systems, able to create large networks of linked and related information including social, business and academic networks. New techniques can analyse these networks accurately and efficiently leading to changes in the way these networks build themselves. This has been demonstrated here through the use of Co-Pilot and CoRank.

Bibliography

- [1] Pritchard, A.: Statistical bibliography or bibliometrics. *Journal of Documentation* **25**(4) (1969) 348–349
- [2] Egghe, L., Rousseau, R.: *Introduction to informetrics: quantitative methods in library, documentation and information science*. Elsevier (1990)
- [3] Price, D.: Networks of scientific papers. *Science* **149** (1965) 510–5
- [4] Bergstrom, C.: Eigenfactor: Measuring the value and prestige of scholarly journals
- [5] Garfield, E.: Citation analysis as a tool in journal evaluation. *Science* **178**(4060) (1972) 471–479
- [6] Kleinberg, J.: Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* **46**(5) (1999) 604–632
- [7] Chakrabarti, S., Dom, B., Gibson, D., Kumar, S., Raghavan, P., Rajagopalan, S., Tomkins, A.: Experiments in topic distillation. *ACM SIGIR Workshop on Hypertext Information Retrieval on the Web* (1998)
- [8] Page, C., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bring order to the web. *Manuscript* (1998) 1997–0072
- [9] Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Computer and ISDN Systems* **30**(1-7) (1998) 107–117
- [10] Aksnes, D., Taxt, R.: Peer reviews and bibliometric indicators: a comparative study at a norwegian university. *Research Evaluation* **13**(1) (2004) 33–41
- [11] Meho, L., Sonnenwald, D.: Citation ranking versus peer evaluation of senior faculty research performance: A case study of kurdish scholarship. *Journal of the American Society for Information Science* **51**(2) (2000) 123–138
- [12] Noruzi, A.: Google scholar: the new generation of citation indexes. *Libri* **55**(4) (2005) 170–180
- [13] : Scopus empowers researchers with new citation tracker. Reed Elsevier Press (2006)
- [14] Giles, C., Bollacker, K., Lawrence, S.: Citeseer: an automatic citation indexing system. *Proceedings of the third ACM conference on Digital libraries* (1998) 89–98
- [15] Moed, H.: Citation analysis of scientific journals and journal impact measures. *Current Science* **89**(12) (2005) 1990–1996
- [16] Brody, T., Harnad, S.: Earlier web usage statistics as predictors of later citation impact. *Journal of the American Society for Information Science and Technology* **57**(8) (2006) 1060–1072
- [17] Small, H.: Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science* **24**(4) (1973) 265–269