Information Extraction from DBLP dataset

Antonio Penta

May 12, 2017

Abstract

In this report, I describe the approaches that I have used to extract relevant information from the DBLP dataset. In particular, I have suggested new methods to compute the "impact on fellows" and "influence on research". A topic analysis is also presented based on the well known Latent Dirichlet Allocation method. The document contains also descriptions of the work-flows and of the scripts, which have been developed for this analysis. The choices have also been driven by the time and technological constraints. I have also suggested in the future work section further improvements for the proposed methodology. All the analysis have applied on the data related to the most cited authors. The analysis shows that ranking researchers based only on citations is not comprehensive enough to understand other phenomena like impact on their fellows and their influence in the research community. In fact, the computed ranks present differences according to which aspect of the analysis we are considering.

1 Problem Statement

The objective of this report is to describe the approaches used to derive the following information:

- the most cited authors,
- their topic of interest,
- their prominent publishers,
- their impact on fellow authors,
- their influence on the their field of research.

2 Data Description

The Proximity DBLP database (from now on named as DBLP) [1] contains information on computer science paper extracted from the DBLP Computer Science Bibliography. The data

[1] https://kdl.cs.umass.edu/display/public/DBLP

is a snapshot of the database as of April 12, 2006. It is an XML file. It includes links from publications to their authors and editors and from papers to the journal, proceedings, or book in which they appear, as well as citation links from one publication to another. The Figure 1 depicted the objects and the links involved in the DBLP database.

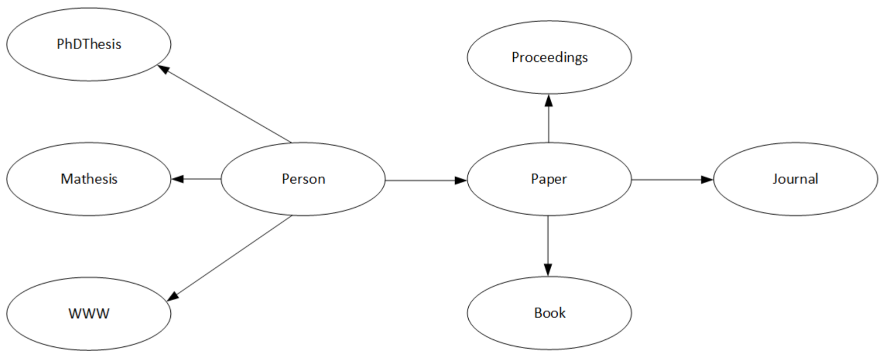

Figure 1: The objects and links in the Proximity DBLP database.

3 Data Extraction

In order to extract the information from the DBLP, I adopt the work-flow explained in Figure 2. The main idea is to extract the relevant information, store them in CSV files, and then importing this content in a relational database[2]. This choice is due to the fact that the proposed work-flow is compatible with the available technological stack and at same time, SQL can be efficiently used to query and combine all the data required for the analysis. An alternative could be the use of graph-based database, that can preserve the original structure of the data, but in the considered database all the required relations can be retrieved or obtained by applying the join operator.

The file are extracted using a python script **scripts/data_extraction.py**, which is parsing the XML file and it is extracting a set of files in CSV format with all the relevant information. The script is navigating the xml tree looking for relevant data and filtering the target links and objects. In order to ensure that the extracted data are correct a set of assertions are used in the code.
The database is generated using the SQL script (**sql/schema.sql**), while the import in the database is done using the script **scripts/db_insert.py**. For the import in the database, I have used the Pandas library[3] that offers a simple method to save dataframe in the database.

I have also used constraints in the relational tables (such as primary key and foreign key) in order to access the validity of the extracted data in terms of uniqueness and containment.

From the original file, I have created the following tables:

- **author**, **paper**, **publisher**, **type** which follows the schema described in Table 1 and they contain the name of the authors, the title of the paper, the name of the publisher

[2] I have used PostgreSQL database for storing all the data.
[3] http://pandas.pydata.org/

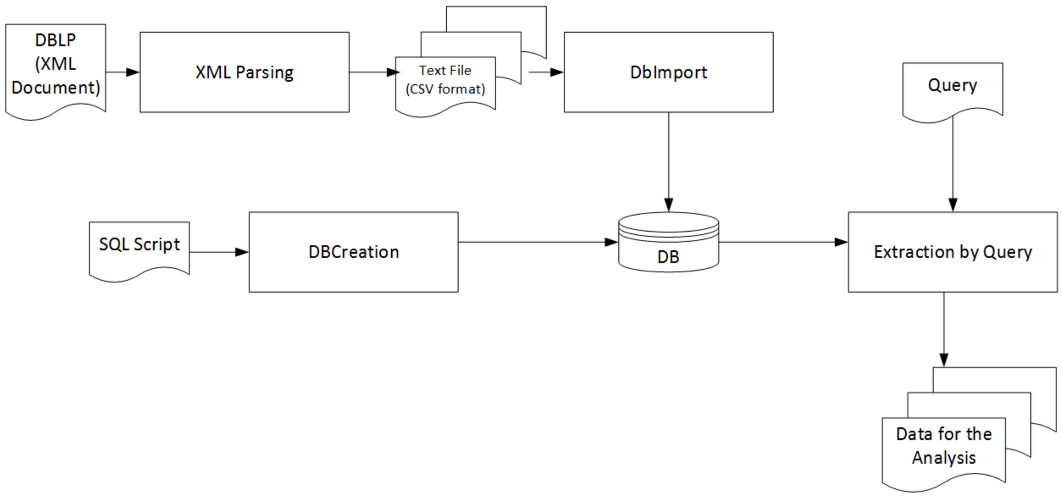

Figure 2: Data Extraction work-flow

Object Relation	
id	**name**

Table 1: Object Relation Schema

and the type of object respectively.

- **author_of**, **cites**, **book**, **journal**, **proceedings**, which follows the schema described in Table 2 and they contain the authorship and citation relations, the association with the book, journal and proceedings respectively.

3.1 Data Check

In order to check the correctness of the extraction process, I have sub-sampled the data and then I manually check the meaning of data by querying the web with the data stored in the tables. This process could be also automated by a script that scrapes the web pages of the authors and it checks if the page with the research or the one with the list of papers matches the data in the database. The idea is to cross-related two different sources in order to check the validity of one by assuming the correctness of the other one. I have randomly sub-sampled the data to avoid this checking operation for all the data which can be too time-consuming. For example, in order to check the authorship data, a small table has been created a with a random selection as follows:

```
create table dblp.random_author_of as
select * from dblp.author_of order by random() limit 5;
```

Then, I have joined the data in order to know the name of the authors for the randomly selected papers, the query is described as as follows:

Link Relation		
id	**id_l**	**id_r**

Table 2: Link Relation Schema

```
select a.id as id_author, a.name as name,
p.id as id_paper, p.name as title
from dblp.author as a ,dblp.random_author_of as l, dblp.paper as p
where a.id=l.id_l and l.id_r=p.id;
```

This is also helps to speed-up the join operation between two tables, whose complexity is $O(NM)$ (worst case), where N and M are the dimensions of the rows in the tables. In the file **doc/db_query.txt**, there are all the queries used in this paper.

4 Analysis

4.1 Most Cited Authors

The most cited authors can be obtained by the above database by running the following queries[4]

- First, I create a table with the number of citations for each paper:

```
create table dblp.number_citation as
select id_r as id,count(*) as number_citations
from dblp.cites group by id_r
order by number_citations desc;
```

- Then, I create a table with the total number of citations for each author:

```
create table dblp.author_citation as
select  a.id as id_author, a.name as name, c.id as id_paper,
c.number_citations as number_citations
from dblp.author as a, dblp.author_of as l, dblp.number_citation as c
where a.id=l.id_l and l.id_r=c.id  order by number_citations desc;
```

- Then, I select the top-50 authors:

```
select id_author as id_author ,name as name_author,
sum(number_citations) as total_number_citations
from  dblp.author_citation  group by (id_author,name)
order by total_number_citations desc limit 50;
```

[4] I can obtain the same results with less queries but I have decided to keep the above steps to improve the readiness of the SQL.

The list of the most cited authors (Top-50) is reported in Table 4 in the Appendix 7. In particular, I have selected the Top-50, in order to ensure the diversity in the research topics and at same time I have enough data that can be easily processed with the available computational resources.

4.2 Topics of Interest

The topic extraction step is done using the well known probabilistic model Latent Dirichlet Allocation(LDA) [1]. The core idea of the LDA is to model the document as a bag of words generated from a mixture of topics. A topic is a distribution of words. By observing the co-occurrences of the words in the input documents, the LDA is inferring the words distribution among the topics, and the distribution of topics within the documents. In the original paper, the inference is done using a Bayesian approach based on variational inference. Since the first seminal paper, many models have been proposed in literature to enhance aspects such as the topic extraction, accuracy, interpretability as well as the inference method.
Recent analysis [2] has shown that LDA is more stable in terms of results compared to other approaches like SVD and NMF. In particular, I have used the implementation of the LDA in the GenSim library [5]. In this case, the LDA method is applied to the collection of documents. Each document is obtained by aggregating all the titles of the papers of a given author. This approach let me to avoid the sparsity problem that I have by considering each title of a paper as one main document. This choice is also due to the fact that one of the main hypothesis of the LDA method is that each document is a mixture of topics, while a title of a paper is most of the time related to just one main topic. In particular, I have focused my analysis only on the most cited authors detected in the previous section.

The work-flow for the topic extraction is depicted in Figure 3.

In particular, with the following query, I have retrieved all the papers written by the Top-50 authors:

```
select top.id_author as id_author,
top.name_author as name_author, p.name as name_paper
from dblp.top_authors as top,
dblp.author_of as l, dblp.paper as p
where top.id_author=l.id_l and l.id_r=p.id;
```

4.2.1 Data Cleaning

In particular, the textual data are tokenized and cleaned using the following function(**function/nlp.py**):

```
def tokenize_and_clean(text):
    if text is not None:
        tokens = list(map(lambda x: wnl.lemmatize(x.lower()),
        nltk.word_tokenize(text)))
        filtered_tokens = [w for w in tokens
                if w not in stopwords.words('english')
```

[5]https://radimrehurek.com/gensim/

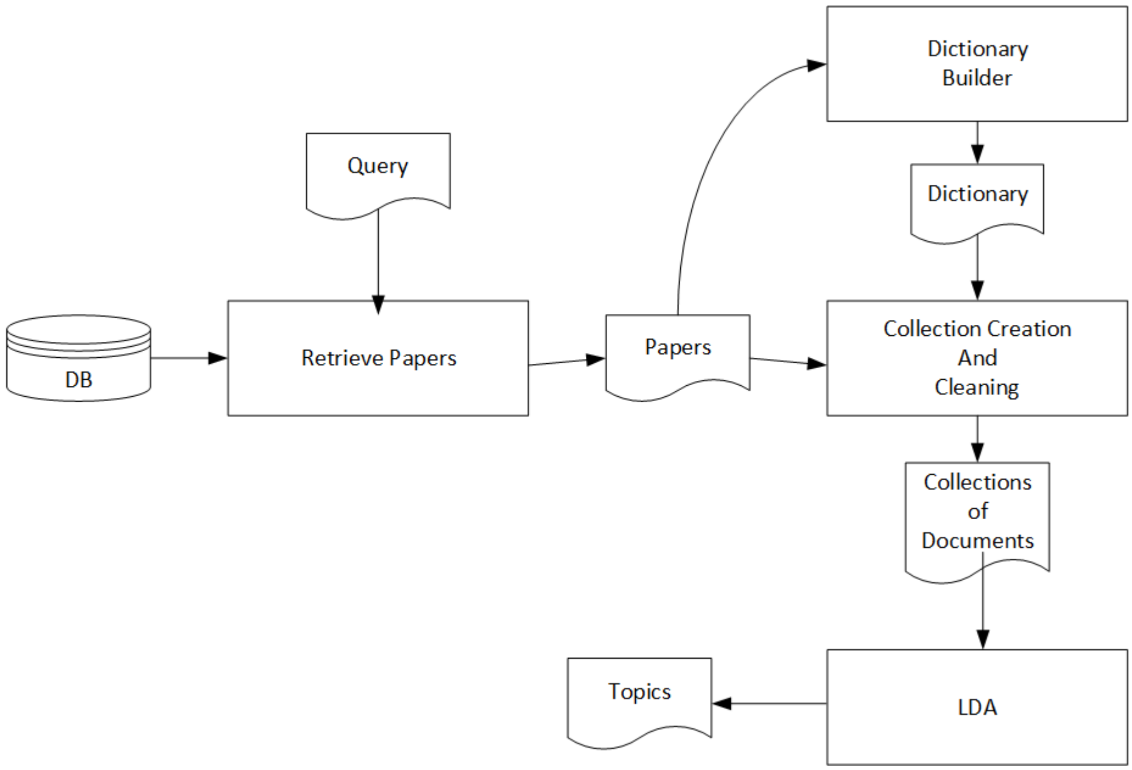

Figure 3: Topic of Interest work-flow

```
                and not w.isnumeric() and
            w not in string.punctuation and len(w)>1]
        return filtered_tokens
    else:
        return ''
```

The above function uses the NLTK[6] library to remove the stop words and to get the lemmatization. In order to remove words not related with any research topics, I have built a domain stop list (**scripts/domain_stop_list.txt**). This process has been done manually, so in order to speed-up this process concentrating the attention only on few terms avoiding the examination of all the 4380 words, I have used the scripts **scripts/create_dict_from_titles.py** and **scripts/extract_idf.py** to rank the terms. In particular, I have focused the attention on the terms with high or low Inverse Document Frequency (IDF). In fact, the IDF is helping me to extract rare and frequent terms within the collection, (IDF of a rare term is high, whereas the IDF of a frequent term is likely to be low.) In Table 5 and 6 in Appendix 7 the Top-50 and Last-50 terms with their IDFs are reported (highest and lowest respectively).

4.2.2 Topic Coherence Analysis

.

[6] http://www.nltk.org/

In the LDA, the main parameters to tune are the number of the topics and the number of iterations. In order to select the best parameters, I have run an experiment related to the Topic Coherence (TC). In this experiment, I have seen how the TC is changing for different values in the numbers of topics and iterations. I have used the TC measure defined in [3], which measures how much a common word triggers a rarer word, this let me to understand if there are meaningless topics among the extracted ones. In particular, I have measured the variance among the Topic Coherence for different number of topics and iterations. The analysis is depicted in Figure 4. According to this experiment 10 topics with an iteration of 20 steps let me to have less variance.[7] The code of this analysis is in **scripts/estimate_topic_lda.py**, while the topics are extracted with **scripts/extract_topic_lda.py**.

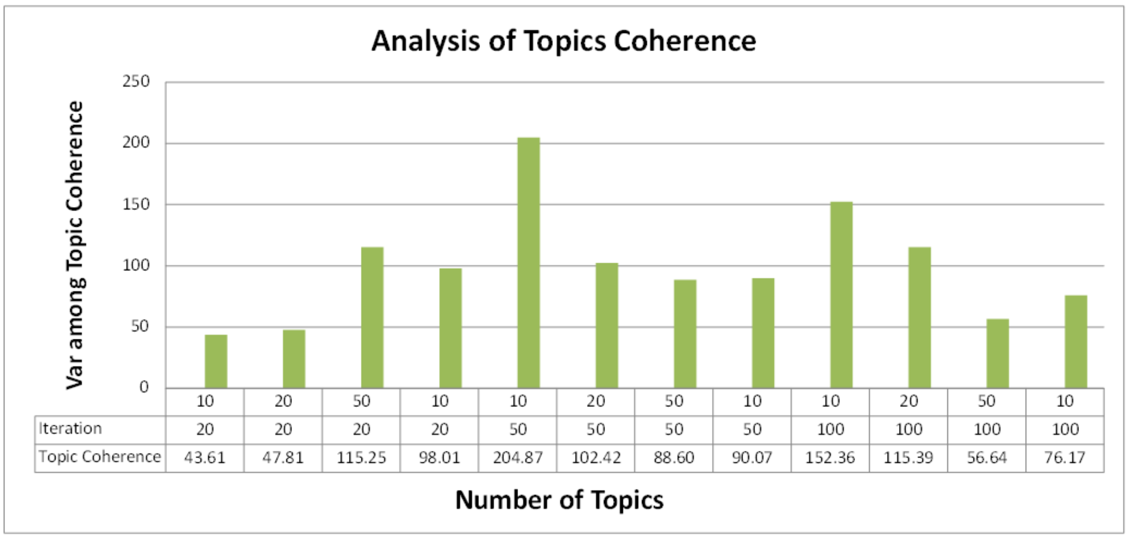

Figure 4: Topic Coherence Analysis

4.2.3 Results

The extracted topics with the Top-10 words are depicted in Table 7 while the assignment between the most cited authors and the topics is depicted in Table 8 in the Appendix 7. Most of the topics are related to the research areas such as database, logic, transaction, system, management and so on, which are the main research topics of the most cited authors. I have also provided in Figure 5 the word-clouds for the Top-5 most cited authors. The word-clouds have been generated using the script **scripts/create_word_cloud.py**[8]. All the images related to the word-clouds are stored in **data/wc**.

4.3 Prominent Publisher

The prominent publishers for each selected author can be obtained by querying the data stored in the database.

[7] A more robust analysis should use a dataset with a collections obtained by different authors.

[8] It uses the library `https://github.com/amueller/word_cloud`

Figure 5: Word-cloud for the Top-5 most cited authors

I have noticed that some journals and proceeding do not have any publisher associated. This check can be done by counting the nulls values using left-join operator as follows:

```
select count(*)
from dblp.journal as j left join dblp.publisher as p
on j.id_r=p.id
where p.name is  not null;

select count(*)
from dblp.journal as j left join dblp.publisher as p
on j.id_r=p.id where p.name is  null;
```

In particular, I have 270221 (236) instances of journal that do not (do) have any publisher, while for the proceedings only 7280 (328200) do not (do) have any publisher. Instead, each book has a publisher associated. The most prominent publishers for each author can be obtained as follows:

- First, I create a table with all the papers:

```
create table  dblp.paper_publisher as
(select proc.id_l as id_paper ,p.name as name_publisher
from dblp.proceedings as proc,dblp.publisher as p
where proc.id_r=p.id
union
select book.id_l as id_paper ,p.name as name_publisher
from dblp.book as book,dblp.publisher as p
where book.id_r=p.id
union
select journal.id_l as id_paper, p.name as name_publisher
from dblp.journal as journal,dblp.publisher as p
where journal.id_r=p.id);
```

- Then, I group and count all the related publishers:

```
select top.name_author,p.name_publisher,count(*) as count
from dblp.top_authors as top, dblp.author_of as l,
dblp.paper_publisher as p
where top.id_author=l.id_l and l.id_r=p.id_paper
group by(top.name_author,p.name_publisher)
having count(*)>1
order by top.name_author,count desc;
```

The results are depicted in Tables 9 in Appendix 7.

4.4 Impact On Fellows

I have interpreted the term fellows of an author has is co-author. The **Impact-on-Fellows** measure (IoF) of an author A is defined as follow:

$$IoF(A) = \frac{1}{|Fellow(A)|} \sum_{x \in Fellow(A)} \frac{1}{TC(x)} \sum_{p \in CoAuthor(x,A)} CIT(p) \tag{1}$$

where $Fellow(A)$ is the set of fellows of A, $TC(x)$ is a function that retrieves the total number of citations of the author x, $CoAuthor(x, A)$ is a function that retrieves the set of papers that x co-authored with A and $CIT(p)$ is a function that retrieves the number of citations of the paper p. The rationale behind this measure is that given an author, I would like to measure (in average) how the total number of citations of the fellow authors is influenced by the work co-authored with him. A bigger value for an author in the "Impact-on-Fellows" means that the fellow authors have a total number of citations that is done to the co-authorships with

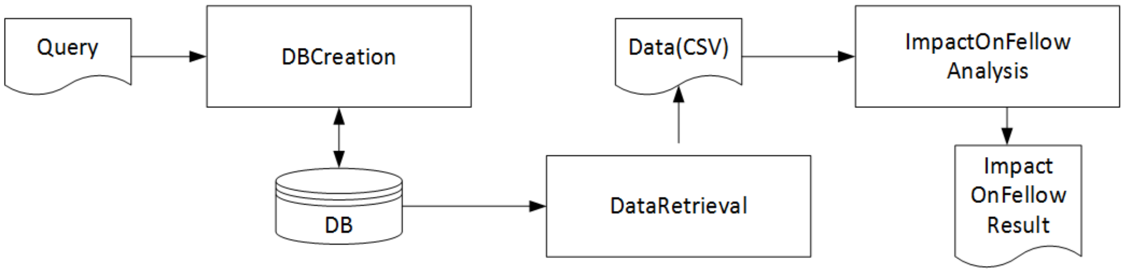

Figure 6: Impact On Fellows work-flow

top_author_id	top_author_name	tot_cit_top_author	fellow_author_id	fellow_author_name	tot_cit_fellow_author	id_paper	paper_cit
728521	Donald D. Chamberlin	978	829382	Raymond A. Lorie	1493	111283	33
728521	Donald D. Chamberlin	978	831713	Donald R. Slutz	78	111283	33
728521	Donald D. Chamberlin	978	832567	W. Frank King III	342	154667	244
728521	Donald D. Chamberlin	978	747876	Morton M. Astrahan	840	154667	244
728521	Donald D. Chamberlin	978	832567	W. Frank King III	342	274401	3
728521	Donald D. Chamberlin	978	833896	Bradford W. Wade	384	111283	33
728521	Donald D. Chamberlin	978	746576	Michael J. Carey	1439	502262	3
728521	Donald D. Chamberlin	978	833900	Gianfranco R. Putzolu	445	154667	244

Table 3: Snapshot of the table used to compute the Impact on Fellow Analysis

him.
In order to do this analysis, I have applied the work-flow described in Figure 6

In particular, I have built a co-authorship relation table, whose a small selection is depicted in Table 3. It stores all the information that I need for this analysis: the total number of citations for each author, their names, the total number of citations for each involved paper.

The process for computing the IoF on the most cited authors is described as follows:

- First, I self-join the table **author_of** to get the co-authorships. I focus my attention only on the most cited authors.[9]:

```
create table dblp.fellow_top_author_1 as
select a1.id_l as top_author,a2.id_l as fellow_author,
a1.id_r as id_paper
from dblp.author_of as a1, dblp.author_of as a2
where a1.id_r=a2.id_r and a1.id_l<>a2.id_l
and a1.id_l in
(select id_author from dblp.top_authors);
```

- Then, I add the names of the most cited authors:

```
create table dblp.fellow_top_author_2 as
select f.top_author as top_author_id,a.name as top_author_name,
f.fellow_author as fellow_author_id, f.id_paper as id_paper
from dblp.fellow_top_author_1 as f, dblp.author as a
where f.top_author=a.id;
```

[9] The last condition let me to reduce the number of rows from 3985638 to 14054, which is more manageable with the considered technological stack.

- Then, I add the names of the fellow authors:

```
create table dblp.fellow_top_author_3 as
select f.top_author_id,f.top_author_name,
f.fellow_author_id,a.name as fellow_author_name,
f.id_paper as id_paper
from dblp.fellow_top_author_2 as f, dblp.author as a
where f.fellow_author_id=a.id;
```

- Then, I create a table for the total number of citations for each author:

```
create table dblp.author_total_citations
as select id_author as id_author ,name as name_author,
sum(number_citations) as total_number_citations
from  dblp.author_citation
group by (id_author,name);
```

- Then, I add the total citations for the most cited authors:

```
create table dblp.fellow_top_author_4 as
select f.top_author_id,f.top_author_name,
a.total_number_citations as citations_top_author,
f.fellow_author_id,f.fellow_author_name ,f.id_paper
from dblp.fellow_top_author_3 as f,
dblp.author_total_citations as a
where f.top_author_id=a.id_author;
```

- Then, I add the total citations for the fellow authors:

```
create table dblp.fellow_top_author_5 as
select f.top_author_id,f.top_author_name,
f.citations_top_author,
f.fellow_author_id,f.fellow_author_name,
a.total_number_citations as citations_fellow_author,
f.id_paper
from dblp.fellow_top_author_4 as f,
dblp.author_total_citations as a
where f.fellow_author_id=a.id_author;
```

- Then, I add the paper citations, and I obtain the final table:

```
create table dblp.fellow_top_author_6 as
select f.top_author_id,f.top_author_name,
f.citations_top_author,
f.fellow_author_id,f.fellow_author_name,
f.citations_fellow_author,f.id_paper,
c.number_citations as paper_citations
```

```
from dblp.fellow_top_author_5 as f,
dblp.number_citation as c
where f.id_paper=c.id;
```

- I run also the following query (expecting an empty result) to check the consistency of the results.

```
select *
from dblp.fellow_top_author_5 as f,
dblp.number_citation as c
where f.id_paper=c.id and
f.citations_top_author<c.number_citations or
f.citations_fellow_author<c.number_citations;
```

The results are depicted in Table 10. The python code implementing the equation 1 is in **scripts/impact_computation.py**. It worth to note that there are authors that have a lot of citations but they did not have a big impact on their fellows maybe due the fact that they have more independent research collaborators.

4.5 Influence on Research

The "Influence on Research" (IoR) is modelled as follows: let us consider the set of all the authors $\mathcal{A}$, a most cited author $a^* \in \mathcal{A}$ and the set of his research topics $\mathcal{T}^*$. We can assume that a^* is influential in his/her research area if the authors in $\mathcal{A} - \{A^*\}$ that have published papers related to $\mathcal{T}^*$ have also cited the work of a^*. In order to express the IoR equation, let us consider the following notation:

- $\mathcal{A}$: the set of all the authors
- $\mathcal{A}^*$: the set of Top-k cited authors.
- $\mathcal{P}$: the set of all the papers written by authors $\mathcal{A} - \mathcal{A}^*$, where each paper has at least one citation to the work of an $a \in \mathcal{A}^*$.
- a^*: an author in $\mathcal{A}^*$.
- $\mathcal{T}^*$: the research topic of a^*.
- PT: paper topic matrix, where $PT[i, j]$ tells us how much the paper $i \in \mathcal{P}$ is related to the topic $j \in \mathcal{T}^*$.
- AT: topic array, where $AT[i]$ measures how much the topic $i \in \mathcal{T}^*$ is related to the research activity of a^*.
- CA: citation array, for each paper in $\mathcal{P}$ we have $CA[i]$ is 1 if the paper i cites a^* otherwise is 0.

Then, the IoR measure for an author a^* is computed as follow:

$$IoR(a^*) = sum(AND(\Phi(PT \cdot \Phi(AT, th), th), CA)) \tag{2}$$

where Φ is function that set to 1 the values of the input matrix/array if the value are greater or equal to the threshold th otherwise 0; AND is the logical "and" between two binary arrays. The main idea of the Equation 2 is to project the papers on the topic space of the considered author trough the *dot* product and then compute an hamming similarity between the selected papers with the citation array. An author is more influential if there are more papers from different authors belonging to his research area that have cited him. In this case, I have applied the LDA model to the title of each paper, and the model is the one learned in Section 4.2.

The work-flow for the IoR measure is depicted in Figure 7

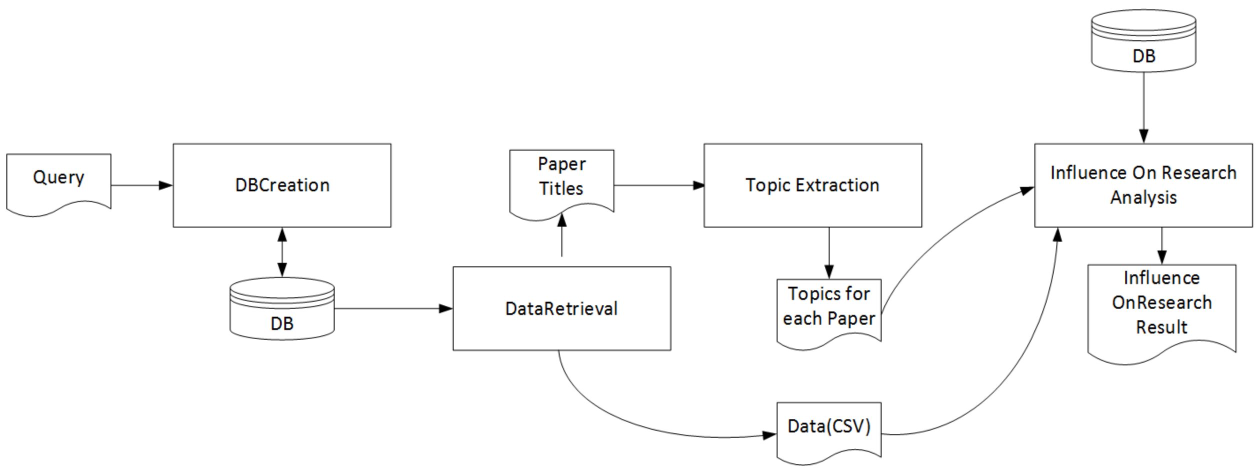

Figure 7: Influence On Research work-flow

The code is in the file **scripts/impact_computation.py**. In particular, I run the following query to generate the data useful for this analysis:

- This query is used to select the papers that have cited the most cited authors.

```
create table dblp.paper_citing_top_author
as select p.id as id_paper, p.name as name_paper,
top.id_author as id_top_author,
top.name_author as top_name_author
from dblp.paper as p, dblp.cites as c,
dblp.author_of as aof,
dblp.top_authors as top
where
p.id=c.id_l
and
aof.id_r=c.id_r
and
```

```
aof.id_l=top.id_author and p.id not in
(select aof2.id_r from dblp.author_of as aof2,
dblp.top_authors as top2
where aof2.id_l=top2.id_author)
```

- This query is used to obtain the title that are processed with the LDA model trained in Section 4.2.

```
select distinct id_paper, name_paper
from dblp.paper_citing_top_author
```

The result of this analysis is depicted in Table 11 in Appendix 7. I have used a threshold of 0.5 for the function Φ.

5 Future Works

Here, I will list some of the improvement that can be done in this analysis as future works:

- The most cited authors are mostly related to the database area, due the fact that the DBLP repository was created originally to host the work for that community. It will be interesting to do the analysis selecting the top-k researchers by research topics rather than only by citations.

- Topic modelling is an active research area, in particular it will be interesting to explore with this data models like Author-Topic model [4][10] or models based on distributed reppresentation such as word embedding models (word2vec and doc2vec have become very popular in text processing [5]).

- It will be also interesting to explore as IoF measure a gravity formulation as follows:

$$IoF(A) = \frac{1}{|Fellow(A)|} \sum_{x \in Fellow(A)} \frac{CIT(A) * CIT(x)}{CoAuthorship(x, A)^2} \tag{3}$$

  where the $CIT()$ is a function that return the number of citations for the input authors and $CoAuthorship(,)$ is a function that returns the number of papers that the input authors have co-authored together.

- For the influence on research, it will be interesting to apply graph-based approach such as discovering hubs and authorities in the citation or co-authorship graphs based on the HITS algorithm [6] and other graph based approaches [7].

[10] http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

6 Conclusions

The analysis shows that ranking researchers based only on citations is not comprehensive enough to understand other phenomena like impact on their fellows and their influence in the research community. In fact, the computed ranks present differences according to which aspect of the analysis we are considering.

7 Appendix

Here, I have reported the results of the analysis.

References

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. *Latent Dirichlet allocation*. Journal Machine Learning Research. 3 (March 2003), 993-1022.

[2] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. *Exploring topic coherence over many models and many topics*. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12). Association for Computational Linguistics, Stroudsburg, PA, USA, 952-961.

[3] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. *Optimizing semantic coherence in topic models*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11). Association for Computational Linguistics, Stroudsburg, PA, USA, 262-272

[4] Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. 2010. *Learning author-topic models from text corpora*. ACM Transaction. Information. System. 28, 1, Article 4 (January 2010), 38 pages.

[5] Mikolov, Tomas; Sutskever, Ilya; Chen, Kai; Corrado, Greg S.; Dean, Jeff *Distributed representations of words and phrases and their compositionality*. Advances in Neural Information Processing Systems.(2013) arXiv:1310.4546

[6] Kleinberg, Jon (1999). *Authoritative sources in a hyperlinked environment* (PDF). Journal of the ACM. 46 (5): 604632.

[7] Malliaros, Fragkiskos D., and Michalis Vazirgiannis. *Clustering and community detection in directed networks: A survey*. Physics Reports 533.4 (2013): 95-142.

Id Author	Name Author	Citations
768838	Jeffrey D. Ullman	3407
779656	Michael Stonebraker	2611
747382	David J. DeWitt	2270
746569	Jim Gray	1912
743701	Philip A. Bernstein	1900
746591	David Maier	1612
741466	Serge Abiteboul	1567
829382	Raymond A. Lorie	1493
728577	E. F. Codd	1465
746576	Michael J. Carey	1439
750731	Won Kim	1392
743700	Nathan Goodman	1358
740157	Hector Garcia-Molina	1332
729520	Yehoshua Sagiv	1288
778747	Catriel Beeri	1272
740263	Rakesh Agrawal	1263
746577	Raghu Ramakrishnan	1102
740400	Umeshwar Dayal	1066
819104	Francois Bancilhon	1065
728521	Donald D. Chamberlin	978
746578	Jennifer Widom	971
746704	Christos Faloutsos	964
798404	Richard Hull	962
729527	Ronald Fagin	949
728727	Shamkant B. Navathe	884
741612	Stefano Ceri	881
747876	Morton M. Astrahan	840
747483	Bruce G. Lindsay	838
729526	Moshe Y. Vardi	836
779750	Jeffrey F. Naughton	828
832759	Irving L. Traiger	824
747484	Hamid Pirahesh	818
738742	C. Mohan	813
771449	Eugene Wong	804
735307	Abraham Silberschatz	795
743881	Peter P. Chen	776
729162	Alberto O. Mendelzon	767
828708	Kapali P. Eswaran	758
733943	Nick Roussopoulos	724
813160	Alfred V. Aho	716
736751	Patrick Valduriez	715
747210	Carlo Zaniolo	704
746571	H. V. Jagadish	692
746601	Yannis E. Ioannidis	683
821255	Goetz Graefe	677
736809	Peter Buneman	662
836266	Stanley B. Zdonik	649
740224	Randy H. Katz	643
743679	Paris C. Kanellakis	642
747343	Hans-Jorg Schek	640

Table 4: Top-50 cited authors.

word	IDF
database	2.42145619533
system	2.8924146088
data	3.02034723097
query	3.29733401431
relational	3.83189798708
management	3.98884587152
model	4.02207151915
distributed	4.15947138856
language	4.19097005561
object	4.27870896992
using	4.32335924366
design	4.40912606542
algorithm	4.46015754543
information	4.46015754543
xml	4.46540690132
transaction	4.56493649666
performance	4.57663253643
approach	4.58253225855
logic	4.59443716106
object-oriented	4.59443716106
application	4.60648549958
web	4.61868077267
processing	4.68855045263
research	4.68855045263
rule	4.74958634322
efficient	4.77078855087
base	4.78517728832
optimization	4.78517728832
view	4.85261856912
mining	4.86824388702
program	4.90840992874
control	4.95884078237
schema	4.97623252508
parallel	4.99393210218
relation	5.03029974635
network	5.04899187936
evaluation	5.05847062332
architecture	5.08745816019
abstract	5.09731045664
concurrency	5.10726078749
dbms	5.10726078749
implementation	5.11731112334
analysis	5.13771999497
active	5.14808278201
querying	5.15855408188
dependency	5.16913619121
problem	5.16913619121
constraint	5.21262130315
join	5.21262130315
method	5.22379460374

Table 5: Words (50) with lowest Inverse Document Frequency

word	IDF
webhouse	9.01928379292
webratio	9.01928379292
webservice	9.01928379292
webviews	9.01928379292
weight	9.01928379292
weighted	9.01928379292
wfms	9.01928379292
where_s	9.01928379292
whisper	9.01928379292
white	9.01928379292
whiteboards	9.01928379292
whitney	9.01928379292
whole	9.01928379292
wi	9.01928379292
wild	9.01928379292
win	9.01928379292
windowed	9.01928379292
winmagic	9.01928379292
winning	9.01928379292
wiring	9.01928379292
wise_01	9.01928379292
wish	9.01928379292
witness	9.01928379292
worklflow	9.01928379292
workplace	9.01928379292
workstation-mainframe	9.01928379292
worldwide	9.01928379292
write-optimized	9.01928379292
writers_	9.01928379292
wsq/dsq	9.01928379292
wxpressions	9.01928379292
x-diff	9.01928379292
xbit	9.01928379292
xml-published	9.01928379292
xml-ql	9.01928379292
xml-sql	9.01928379292
xml-to-relational	9.01928379292
xmltm	9.01928379292
xpath	9.01928379292
xpref	9.01928379292
xqery	9.01928379292
xrm	9.01928379292
xroaster	9.01928379292
xsearch	9.01928379292
xyleme	9.01928379292
yappers	9.01928379292
you_re	9.01928379292
youserv	9.01928379292
zoom	9.01928379292
zoomable	9.01928379292
zooming	9.01928379292

Table 6: Words (50) with highest Inverse Document Frequency

Topic Index	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10
0	database 0.0291	dependency 0.0187	relational 0.0176	system 0.0142	information 0.0141	theory 0.0135	expression 0.0122	common 0.0121	scheme 0.0113	knowledge 0.0111
1	database 0.0006	data 0.0006	query 0.0005	system 0.0005	model 0.0005	object 0.0005	language 0.0005	web 0.0005	information 0.0005	relational 0.0005
2	database 0.0443	query 0.0363	xml 0.0347	web 0.0211	data 0.0160	querying 0.0137	review 0.0114	dependency 0.0103	exodus 0.0092	relational 0.0089
3	database 0.0581	data 0.0402	query 0.0282	language 0.0209	mining 0.0208	system 0.0178	logic 0.0155	deductive 0.0149	program 0.0130	rule 0.0124
4	design 0.0279	database 0.0229	network 0.0196	service 0.0177	system 0.0175	performance 0.0174	using 0.0102	wireless 0.0101	application 0.0097	architecture 0.0093
5	database 0.0565	query 0.0357	data 0.0253	design 0.0155	xml 0.0154	system 0.0153	language 0.0151	web 0.0131	object 0.0119	application 0.0112
6	logic 0.0401	query 0.0313	database 0.0286	program 0.0222	model 0.0199	dependency 0.0166	data 0.0141	problem 0.0140	language 0.0136	checking 0.0128
7	database 0.0614	system 0.0541	data 0.0367	management 0.0187	query 0.0162	relational 0.0157	distributed 0.0151	transaction 0.0120	model 0.0115	object 0.0099
8	database 0.0007	system 0.0006	data 0.0006	query 0.0006	management 0.0005	model 0.0005	approach 0.0005	language 0.0005	relational 0.0005	application 0.0005
9	data 0.0342	mining 0.0304	using 0.0255	database 0.0170	fast 0.0164	method 0.0157	performance 0.0145	indexing 0.0140	fractal 0.0119	image 0.0115

Table 7: Top-10 words for each topic. Each value represents how much likely a word is assigned to the topic.

Name Author	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9
Bruce G. Lindsay	0.4158	0	0	0	0	0	0	0.5835	0	0
Michael J. Carey	0	0	0	0	0	0	0	0	0	0.9997
Hans-Jorg Schek	0.9996	0	0	0	0	0	0	0	0	0
Hector Garcia-Molina	0	0	0	0	0	0	0	0	0	0.9998
Yehoshua Sagiv	0	0	0	0.9995	0	0	0	0	0	0
Francois Bancilhon	0.1587	0	0	0	0	0	0	0	0.6891	0.1514
Umeshwar Dayal	0.8510	0	0	0	0.1487	0	0	0	0	0
Kapali P. Eswaran	0.7188	0	0	0	0	0.2775	0	0	0	0
Rakesh Agrawal	0	0	0.9996	0	0	0	0	0	0	0
Donald D. Chamberlin	0.9988	0	0	0	0	0	0	0	0	0
Paris C. Kanellakis	0	0	0	0	0	0	0	0	0.9993	0
Richard Hull	0	0	0	0	0.2843	0	0	0	0.7153	0
C. Mohan	0	0	0	0	0	0	0	0.9995	0	0
Jim Gray	0.8369	0	0	0	0	0	0	0	0	0.1628
Alfred V. Aho	0	0	0	0.9991	0	0	0	0	0	0
Shamkant B. Navathe	0	0	0	0	0.9997	0	0	0	0	0
Raymond A. Lorie	0.6203	0	0	0	0	0	0	0	0	0.3787
Catriel Beeri	0	0	0	0	0	0	0	0	0.9994	0
Goetz Graefe	0	0	0.9989	0	0	0	0	0	0	0
Abraham Silberschatz	0	0	0	0	0	0	0	0	0	0.9997
Moshe Y. Vardi	0	0	0	0.7279	0	0	0	0	0.2719	0
Ronald Fagin	0	0	0	0	0	0	0	0	0.9995	0
Eugene Wong	0.9985	0	0	0	0	0	0	0	0	0
Peter Buneman	0	0	0	0	0.9992	0	0	0	0	0
Alberto O. Mendelzon	0	0	0	0	0	0.9915	0	0	0	0
Irving L. Traiger	0.9978	0	0	0	0	0	0	0	0	0
Jennifer Widom	0	0	0	0	0.9996	0	0	0	0	0
Christos Faloutsos	0	0	0	0	0	0	0.9997	0	0	0
Hamid Pirahesh	0	0	0	0	0	0	0	0.9992	0	0
Michael Stonebraker	0.8847	0	0	0	0.1151	0	0	0	0	0
Stanley B. Zdonik	0	0	0	0	0.8519	0	0	0	0	0.1476
Serge Abiteboul	0	0	0	0	0.4246	0	0	0.2049	0.3702	0
Peter P. Chen	0	0	0	0	0	0.9986	0	0	0	0
Yannis E. Ioannidis	0	0	0	0.0931	0.2462	0	0.3035	0	0	0.3569
Stefano Ceri	0	0	0	0	0.9997	0	0	0	0	0
E. F. Codd	0.5673	0	0	0	0	0	0	0	0.4316	0
Jeffrey F. Naughton	0	0	0.4920	0.1151	0	0	0	0	0	0.3926
Won Kim	0.0637	0	0	0	0.0848	0	0	0	0	0.8512
Randy H. Katz	0	0	0	0	0	0	0	0	0	0.9997
H. V. Jagadish	0	0	0	0	0	0	0.9996	0	0	0
Jeffrey D. Ullman	0	0	0	0.9997	0	0	0	0	0	0
Raghu Ramakrishnan	0	0.6212	0	0	0	0	0	0	0.3784	0
Carlo Zaniolo	0	0	0	0	0	0	0	0.6826	0.3171	0
Patrick Valduriez	0	0	0	0	0	0	0	0	0	0.9996
David Maier	0	0	0	0.0670	0.8311	0	0	0	0.1016	0
David J. DeWitt	0	0	0	0	0	0	0	0	0	0.9997
Morton M. Astrahan	0.8543	0	0.0518	0	0	0	0	0	0	0.0925
Nick Roussopoulos	0	0	0	0	0	0.9995	0	0	0	0
Nathan Goodman	0	0	0	0	0	0	0	0	0.0582	0.9413
Philip A. Bernstein	0.1253	0	0	0	0	0	0	0	0.1830	0.6915

Table 8: Topic assignment for each author. Each value represents how much likely is the topic assigned to the author.

Author	Most Prominent Publisher
Won Kim	IEEE Computer Society(11)-Springer(10)-ACM Press(9)-Morgan Kaufmann(8)
Yehoshua Sagiv	ACM(18)-ACM Press(17)-Springer(16)-Morgan Kaufmann(4)-IEEE Computer Society(3)
Serge Abiteboul	Springer(30)-ACM Press(19)-ACM(17)-Morgan Kaufmann(12)-INRIA(10)-IEEE Computer Society(6)
H. V. Jagadish	Morgan Kaufmann(25)-ACM Press(16)-Springer(16)-ACM(14)-IEEE Computer Society(13)
Christos Faloutsos	ACM(25)-IEEE Computer Society(23)-Morgan Kaufmann(20)-Springer(16)-ACM Press(13)
Raymond A. Lorie	ACM(5)-Morgan Kaufmann(3)-ACM Press(2)-IBM Cambridge Scientific Center(2)-Springer(2)
Shamkant B. Navathe	IEEE Computer Society(25)-Springer(17)-Morgan Kaufmann(9)-ACM(6)-North-Holland(5) ER Institute(5)-ACM Press(3)-CSREA Press(2)-Lawrence Berkeley Laboratory(2)
Nick Roussopoulos	ACM(10)-Springer(10)-Morgan Kaufmann(10)-ACM Press(9)-IEEE Computer Society(7)-North-Holland(4)-CSREA Press(2)
Michael Stonebraker	IEEE Computer Society(29)-ACM Press(18)-Morgan Kaufmann(17)-ACM(13)-Springer(3)
Stefano Ceri	Springer(29)-Morgan Kaufmann(11)-IEEE Computer Society(9)-ACM(7)-ACM Press(6)-Mediterranean Press (2)
Donald D. Chamberlin	ACM(7)-IEEE Computer Society(3)-Springer(2)-Morgan Kaufmann(2)
Hamid Pirahesh	ACM Press(15)-ACM(12)-IEEE Computer Society(10)-Morgan Kaufmann(7)-Springer(3)
Rakesh Agrawal	Morgan Kaufmann(24)-IEEE Computer Society(22)-ACM Press(16)-Springer(15)-ACM(12)
Nathan Goodman	ACM(8)-IEEE Computer Society(5)-Morgan Kaufmann(4)-ACM Press(4)-AAAI(2)
Jeffrey D. Ullman	ACM(30)-ACM Press(19)-IEEE(17)-Springer(12)-IEEE Computer Society(5)-Morgan Kaufmann(3)
Hector Garcia-Molina	IEEE Computer Society(42)-ACM(29)-Springer(28)-Morgan Kaufmann(24)-ACM Press(22)
Moshe Y. Vardi	Springer(99)-ACM(28)-IEEE Computer Society(15)-ACM Press(14)-Morgan Kaufmann(10)-IEEE(5)-Chapman a Hall(2)
Goetz Graefe	Morgan Kaufmann(8)-ACM Press(8)-IEEE Computer Society(7)-GI(2)
David Maier	IEEE Computer Society(16)-Springer(12)-ACM Press(12)-ACM(11)-Morgan Kaufmann(7)-Springer and British Computer Society(2)
Paris C. Kanellakis	Springer(10)-ACM Press(10)-ACM(9)
E. F. Codd	ACM(5)-ACM Press(2)-IBM Cambridge Scientific Center(2)
Jeffrey F. Naughton	ACM Press(20)-Morgan Kaufmann(18)-IEEE Computer Society(16)-ACM(15)-Springer(8)
Alberto O. Mendelzon	ACM(17)-Springer(14)-ACM Press(14)-IEEE Computer Society(8)-Morgan Kaufmann(5)-North-Holland(2)
Peter P. Chen	North-Holland(10)-Springer(7)-ACM(4)-IEEE Computer Society(3)-IEEE Computer Society and North-Holland(2)
Jennifer Widom	ACM(18)-Morgan Kaufmann(15)-ACM Press(12)-Springer(11)-IEEE Computer Society(8)
Michael J. Carey	ACM Press(33)-Morgan Kaufmann(24)-Springer(8)-IEEE Computer Society(6)-ACM(4)
Philip A. Bernstein	ACM(11)-IEEE Computer Society(9)-Morgan Kaufmann(8)-ACM Press(6)-Springer(6)
Eugene Wong	Springer(3)-ACM Press(3)-Morgan Kaufmann(2)-ACM(2)
Stanley B. Zdonik	IEEE Computer Society(12)-Morgan Kaufmann(11)-Springer(9)-ACM Press(9)-ACM(4)
Catriel Beeri	Springer(17)-ACM(11)-ACM Press(7)-Morgan Kaufmann(3)-IEEE(2)-IEEE Computer Society(2)
Hans-Jorg Schek	Springer(33)-IEEE Computer Society(11)-Morgan Kaufmann(10)-ACM Press(6)-ACM(6)-CEUR-WS.org(3)-GI(2)-Kluwer(2)
Jim Gray	ACM Press(12)-ACM(7)-Morgan Kaufmann(6)-IEEE Computer Society(4)-Springer(4)-IBM Cambridge Scientific Center(2)
Randy H. Katz	ACM(12)-IEEE Computer Society(9)-Springer(9)-ACM Press(8)-USENIX(4)-North-Holland(2)-Morgan Kaufmann(2)
Carlo Zaniolo	Springer(29)-IEEE Computer Society(9)-Morgan Kaufmann(7)-ACM Press(7)-ACM(7)
Alfred V. Aho	ACM(9)-IEEE(8)
Raghu Ramakrishnan	ACM Press(22)-Morgan Kaufmann(19)-ACM(18)-IEEE Computer Society(9)-Springer(6)
Peter Buneman	Springer(16)-ACM Press(10)-ACM(5)-Morgan Kaufmann(4)-IEEE Computer Society(4)
Abraham Silberschatz	ACM Press(20)-Morgan Kaufmann(19)-IEEE Computer Society(12)-ACM(8)-USENIX(4)-Springer(3)
Richard Hull	Springer(14)-ACM(12)-ACM Press(11)-IEEE Computer Society(7)-Morgan Kaufmann(6)-CEUR-WS.org(3)
Umeshwar Dayal	IEEE Computer Society(21)-Springer(21)-Morgan Kaufmann(17)-ACM Press(10)-ACM(8)-North-Holland(2)
Morton M. Astrahan	ACM(5)
Irving L. Traiger	IBM Cambridge Scientific Center(3)
Yannis E. Ioannidis	Morgan Kaufmann(18)-ACM Press(16)-Springer(9)-IEEE Computer Society(8)-ACM(5)
Francois Bancilhon	Springer(7)-ACM(6)-Morgan Kaufmann(6)-ACM Press(4)-IEEE Computer Society(2)-INRIA(2)
David J. DeWitt	Morgan Kaufmann(25)-ACM Press(24)-IEEE Computer Society(14)-ACM(12)-Springer(3)
Patrick Valduriez	Morgan Kaufmann(15)-Springer(12)-IEEE Computer Society(10)-ACM Press(5)-INRIA(5)
C. Mohan	ACM Press(13)-Morgan Kaufmann(11)-Springer(9)-IEEE Computer Society(9)-ACM(6)
Bruce G. Lindsay	Morgan Kaufmann(8)-ACM Press(6)-IEEE Computer Society(6)-ACM(3)-IBM Cambridge Scientific Center(2)-Springer(2)
Ronald Fagin	ACM(16)-Springer(8)-Morgan Kaufmann(5)-IEEE Computer Society(3)-ACM Press(3)-IEEE(2)

Table 9: Most Prominent Publishers for the Top-50 Authors.

Author	TotalCitations	ImpactOnFellow
Nick Roussopoulos	724	0.712
Raymond A. Lorie	1493	0.6463
Shamkant B. Navathe	884	0.6428
Richard Hull	962	0.6303
Goetz Graefe	677	0.5598
Irving L. Traiger	824	0.5594
Michael Stonebraker	2611	0.5572
Won Kim	1392	0.5482
Morton M. Astrahan	840	0.5402
Randy H. Katz	643	0.5394
Stanley B. Zdonik	649	0.5334
Donald D. Chamberlin	978	0.5308
David J. DeWitt	2270	0.5294
Nathan Goodman	1358	0.5211
Kapali P. Eswaran	758	0.5207
Peter Buneman	662	0.5076
Patrick Valduriez	715	0.5028
Christos Faloutsos	964	0.4975
Jeffrey F. Naughton	828	0.4966
Eugene Wong	804	0.4814
C. Mohan	813	0.472
Alberto O. Mendelzon	767	0.4717
Hector Garcia-Molina	1332	0.4582
Jim Gray	1912	0.4568
Michael J. Carey	1439	0.4466
Carlo Zaniolo	704	0.446
Abraham Silberschatz	795	0.4417
Peter P. Chen	776	0.4341
Jennifer Widom	971	0.4153
Raghu Ramakrishnan	1102	0.4069
Alfred V. Aho	716	0.4034
Philip A. Bernstein	1900	0.4008
Yannis E. Ioannidis	683	0.3881
Paris C. Kanellakis	642	0.3875
Rakesh Agrawal	1263	0.3832
Hamid Pirahesh	818	0.378
Stefano Ceri	881	0.377
David Maier	1612	0.3632
Serge Abiteboul	1567	0.3581
Yehoshua Sagiv	1288	0.3578
Jeffrey D. Ullman	3407	0.3501
Hans-Jorg Schek	640	0.3483
Ronald Fagin	949	0.3398
Francois Bancilhon	1065	0.3112
Bruce G. Lindsay	838	0.3017
H. V. Jagadish	692	0.2737
Umeshwar Dayal	1066	0.2693
Catriel Beeri	1272	0.262
Moshe Y. Vardi	836	0.1743
E. F. Codd	1465	0.0827

Table 10: Results on the Impact on Fellow Analysis

Author	InfluenceOnResearch
Moshe Y. Vardi	0.9254
Carlo Zaniolo	0.9219
Alfred V. Aho	0.9162
Catriel Beeri	0.8975
Paris C. Kanellakis	0.8939
Peter P. Chen	0.8874
Yehoshua Sagiv	0.8851
Christos Faloutsos	0.8216
Nathan Goodman	0.8083
Hector Garcia-Molina	0.8037
Goetz Graefe	0.8014
Rakesh Agrawal	0.8005
Yannis E. Ioannidis	0.7995
Jeffrey F. Naughton	0.7932
Patrick Valduriez	0.7911
C. Mohan	0.7845
Eugene Wong	0.7784
Peter Buneman	0.7778
Jennifer Widom	0.7732
Abraham Silberschatz	0.7732
H. V. Jagadish	0.7721
Ronald Fagin	0.7719
Bruce G. Lindsay	0.7719
Raghu Ramakrishnan	0.7696
David J. DeWitt	0.769
Hamid Pirahesh	0.769
Richard Hull	0.7677
Morton M. Astrahan	0.7677
Philip A. Bernstein	0.7638
Randy H. Katz	0.7629
Irving L. Traiger	0.7623
Michael J. Carey	0.76
Kapali P. Eswaran	0.7596
Hans-Jorg Schek	0.759
Alberto O. Mendelzon	0.7581
Won Kim	0.7575
Serge Abiteboul	0.7569
Stanley B. Zdonik	0.7559
Jeffrey D. Ullman	0.7535
Francois Bancilhon	0.7517
Shamkant B. Navathe	0.7502
Stefano Ceri	0.7492
Nick Roussopoulos	0.7373
E. F. Codd	0.7356
Umeshwar Dayal	0.7235
David Maier	0.7214
Donald D. Chamberlin	0.7139
Raymond A. Lorie	0.7114
Jim Gray	0.6993
Michael Stonebraker	0.6792

Table 11: Results on the Influence on Research Analysis