Academic Advancement Advice: Author Articles as A. A.

A. A.
(tom7.org foundation) *

1 April 2018

Keywords: A A A A A A

1 Introduction

Those with names falling in the dead-middle of the alphabet—like as a random example, Murphy—have suspected since early life a disadvantage. For example, when the teacher lines up the students for pie, in order to be completely fair and avoid bias, orders them by the perfectly objective criterion of alphabetical order. Occasionally, the teacher realizes that the same students always get pie first, and corrects this bias by using *reverse* alphabetical order. Sometimes the teacher uses alphabetical or reverse-alphabetical by first name, or sometimes last name. All variations of this idea are of course unfair, as is clear to first- and second-graders, who are powerless to rebel without (a) status in the K-12 hierarchy (b) the statistical language in which to formalize their objection and (c) access to a prestigious conference with permissive publication standards.

This paper demonstrates that even within academic paper publishing (which one could argue is obsessed with bias, prestigious conferences, statistical language for formalizing objections, and status in the K-12 hierarchy), this unfairness persists. Specifically, we find that authors whose names come alphabetically earlier have higher citation rates. We also study some related subjects like, "What is the most uncitable paper?"

2 The Data

I obtained a FISA warrant based solely on a salacious and unverified memo funded by SIGBOVIK's political opponents. The warrant was for the AMiner Open Academic Graph [1], which consists of summary information for 154,771,162 papers in about 130 gigabytes of JSON data. Then I performed computer science for longer than it should have taken.

2.1 Relatable Computer Science Tale

This subsection contains a relatable tale of computer science. It can be safely skipped, like all content in this paper.

The task is apparently simple: Parse 130 gigabytes of JSON data to extract the author names and citations from the papers and count them. To begin my journey, I selected a C++ JSON library from 39 alternatives listed on a benchmark page.[2] The first criterion I ordered by was "correctness," for which only two parsers achieved the highest correctness score, 100%. I selected the slower (i.e., more methodical) of the two parsers, since I did not want to do shoddy work.

Next I spent a nice Saturday morning integrating this library into my private build environment, making sure its open source license was properly declared and so on, while I waited for the 130 gigabytes of JSON to download.[1] Miraculously, this was straightforward. I was able to finish writing the code to process the articles before they even finished downloading. However, upon unzipping, I found that each file contained one million articles, not as a single JSON array but as a series of adjacent objects. AdJSONt objects. The JSON library allowed for reading JSON from files, but *not* e.g. for reading a single record from a file pointer repeatedly. So, I needed to do my own ad hoc parsing before feeding each record to the JSON library as strings. OK. Now when I run it, it prints

```
[Segmentation fault
```

*Copyright © 2018 the Regents of the Wikiplia Foundation. Appears in SIGBOVIK 2018 with the abecedarian advantage of the Association for Aomputational Aeresy; *IEEEEEE!* press, Verlag–Verlag volume no. `std::numeric_limits<float>::lowest()`. 1 إد.

[1] I assure the reader that I have the most premium optical-fiber-based Internet. However, the Microsoft Content Distribution Network throttled these downloads mightily, presumably because of Net Neutrality.

This is not my first segmentation fault—let me tell you!—so I knew what to do. I debugged the program, adding assertions using a `CHECK` macro that aborts if its condition is violated. I eventually determined that the bug must be inside the JSON library itself, since the segmentation fault happened between two `CHECK`s where the only code being run was JSON library code. So, deciding that perhaps its 100% correctness benchmark score was overstated, I downloaded and repeated the process with the other, faster (i.e., more slapdash) parser. Of course this parser has a different API so I needed to rewrite my program. (The program is very simple so this was the fastest step; however, getting the library properly incorporated and compiling and then learning its API still required me to apply fundamental principles from my advanced computer science degree.) Once my program was rid of the old crashy library, I ran it, and it produced this result

```
[Segmentation fault
```

Since these two libraries produced the same result, either segmentation violation must be part of the JSON specification, or the bug is instead elsewhere in the program. After reducing the program to a fairly simple test case, I determined that it must be some kind of “Heisenbug” whereby an earlier piece of code was corrupting the heap or similar, causing a later crash only when present. This was not my first Heisenbug—let me tell you!—so I knew what to do. However, not being able to use Valgrind on Windows (did I mention that I'm using Cygwin and mingw64 on the eponymous Windows 7 by the way? Why am I doing that?), I booted Linux in a virtual machine, installed the 87 critical security updates that had been released since I last booted this virtual machine a few months ago, and compiled my program there. I then needed to move one of the JSON files to the virtual computer to test, without having to download 130 GB from the internet again, but the file was apparently too large to transfer using the mysterious “Drag and drop to virtual machine” functionality. I recalled that VirtualBox can mount a shared folder from the host machine if you edit the `/etc/fstab` file to contain the correct code words, and that I had performed this ritual before, but then had commented-out that line in “rescue mode” because I also learned that having this line present during the boot process causes a kernel panic. So I temporarily reinstated the line and mounted the drive and copied the gigantic file into the virtual machine and then removed the dangerous magic from the f-stab for f-safety.

Next I ran Valgrind on this minimal test case and it worked as expected with no unclean behavior detected and no crashing.

Since Linux was no help debugging, I returned to Windows 7 and tried single-step debugging. Here I discovered that the crash in my minimal test case was occurring when using `cerr`. Specifically, the second thing sent to `cerr` would produce a segmentation fault. I then recalled that the `CHECK` macro uses `cerr` and now saw that the `[` in `[Segmentation fault` was probative, for this was the location of the original crash:

```
std::cerr << "[" << pretty_date_.HumanDate() << "] "
          << file << ":" << line << ": ";
```

First I checked for overfull hbox, but this was not the problem. Of course, inside `HumanDate()` I found a hasty patch I had installed in A.D. 2014 called “make logging not spit garbage on mingw,” suggesting that I had encountered and misdiagnosed this issue in the past. Although this led me on a minor herring (e.g. is `__MINGW32__` defined when cross-compiling for x86_64?) HumanDate just returned `"mingw"`, so this really was a problem with `std::cerr` crashing on the second thing printed with it (i.e., with the returned `ostream`).

Minimizing the test case, this bug only occurred after reading a large file all in one syscall (for “performance”, even though doing these huge reads basically locks up my machine). Specifically the file was 2,105,319,548 bytes, which is 98% of 2^{31}, suggesting the possibility of integer overflow in some buffer-sizing code in the weird Cygwin or mingw wrappers for `fread`, `fstat`, etc. This could plausibly cause corruption of the `cerr` ostream, which is probably part of the same code.

Preparing to at least make a bug report if not try to fix it, I upgraded my compiler and runtime from GCC version 5.1.0 to 6.4.0, and the problem completely went away. Since Valgrind also agreed that there was no issue, I chalked this one up to simply an internal C runtime bug that someone else found and fixed in the last 2 years. Thanks!

With `cerr` fixed, I could now easily see the original “bug”, which was a violated assumption: I expected (and `CHECK`ed), optimistically, that if the article had an `author` field then the author would be a string. I accounted for a missing author, an author that was the empty string, but not `author: null`. Thanks!

Finally, I could run the author and citation tallier on all of the JSON files. However, two of the files appeared to be empty. Although this was only a small percentage of the articles, I did not want to skip any because it's possible that they are arranged in a non-random order (e.g. alphabetically!) and that this would bias the results. I learned that although the AMiner database is helpfully split into files each containing one million articles, for ease of handling, two of these files are nonetheless slightly larger than 2^{31} bytes. Such files cannot be cleanly sized with `fstat` (the `st_size` field has type `off_t`, a signed 32-bit type in POSIX) and even if you follow the standard advice of just treating it as unsigned anyway, `fread` will simply fail when asked to read such a large chunk[2] despite the fact that it takes `size_t`, an unsigned type. (?!) So, another afternoon spent doing Computer Science. But all you need to do of course is perform multiple sequential reads.

Finally, the trivial task of counting the number of articles and citations per author could be completed.

3 The Data

This citation database is very noisy. About 1.6% of articles have no authors, and another 1% have a blank (or `null`) author. I verified the comprehensiveness of the dataset by finding my own author record(s)[3] and 275 citations (**Yeah!!**). I also spot-checked that the database includes prestigious conferences, which it does, at least including the "article" *A Record of the Proceedings of SIGBOVIK 2008*, with author *Pennsylvania USA*. It was necessary to normalize author names to remove punctuation, weird unicode stuff like non-breaking spaces, javascript-encoded newlines and nul-bytes, and so on. There are many strange authors in the database, such as

- capinha para celular horizonte artificial iphone (no citations)
- +0aaaaaababla aaaaaea&#xa (no citations)
- a h poop (4 citations)
- john mcm anus[4] (2 citations)
- coffee hour (0 citations)
- nominate com registering dad domains since (0 citations)
- a a (195 citations)
- a a a a m islam (6 citations)

On the other hand, from this hand-picked sample we can already see the citation advantage of having a name that's alphabetically early. Some authors seem to have already intuited this fact.

4 Author Analysis

Author names appear in both "Firstname Lastname" order and "Lastname Firstname" (when this grammar is even applicable). Therefore, I analyzed alphabetizing from front to back (by space-separated token) as well as back to front. I also rejected a large number entries when I could not place the name in alphabetical order because its first character was not ASCII. Later non-ASCII characters are OK, and are just sorted by their UTF-8 encodings (or whatever garbage is in the file). This of course produces a bias (excluding authors who write their names in e.g. Cyrillic and CJK scripts), although it's not clear how to assess the alphabetical hypothesis for them.

To visualize the data, I produced a cumulative distribution function (CDF) for both articles and citations. Considering each author in alphabetical order, I keep a running total of how many articles and how many citations have been seen so far. The x-coordinate is the rank of the author (its index in the alphabetical list) and the y-coordinates are the fraction of articles (or citations) seen so far. The CDFs for the forward and backward token orders are in Figure 1.

These CDFs show that—as expected!—there is a bias towards citing authors alphabetically early, whether we order alphabetically by "first name" or "last name."

There are multiple hypothesized causes:

- When writing a last-minute "related work" section, authors sometimes find papers to cite in alphabetical order, for example, by reading another paper's bibliography, which is sometimes alphabetized.

[2]Curious why I am even using `fstat` and `fread` like some dramatically bearded guy from the 1980s? I have my own library routine for reading a file into a string. When working with large files it's useful to avoid copying, because the contents of the file may be a significant fraction of all RAM—copying is not only expensive, but might temporarily exceed the available RAM. So, you need to allocate the buffer ahead of time and avoid resizing it (which often also performs a copy). It seems like something that everyone would want to do, but it's ridiculously hard to do portably in C or C++. Specifically, there seems to be no way to actually know the size of a file so that you can size the buffer ahead of time. For example, `fseek` to `SEEK_END` and `ftell` does *not* work (it has *undefined behavior*). The `st_size` field from `stat` also does not give correct results e.g. when reading from the `/proc` filesystem. My routine attempts to guess the size of the file and perform a single read in the case that the guess is correct, while still usually avoiding copying or needless work in the case that it's an under- or over-estimate.

[3]This paper is published under the pseudonym A. A., standing for Awesome Author, obviously in order to increase its citation rate. Normally I publish as Tom Murphy VII, which can be found in citations with many broken variations, such as "Vii, T."

[4]Presumably a hostile citation of John McManus.

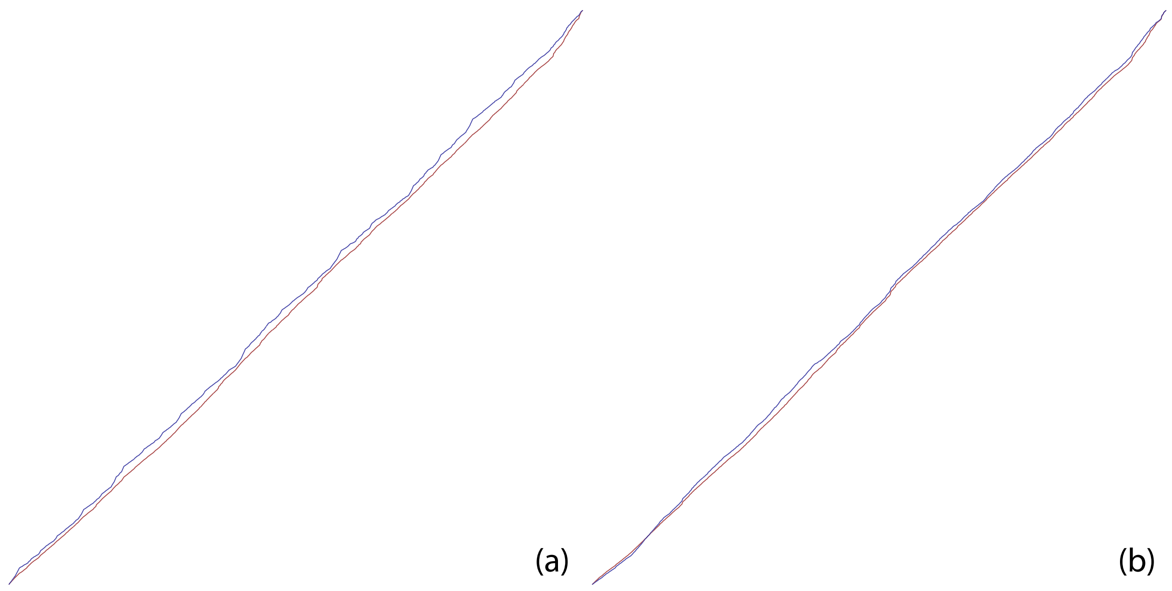

Figure 1: Author CDFs for (a) forward token order and (b) backward token order. (For example, (a) treats `tom murphy vii` as coming between `tom marphy` and `tom myrphie vxx` and (b) treats it as coming between `zzz murphx vii` and `a viii`.) The x-axis is the author's position alphabetically among all authors. The blue lines are the fraction of all citations seen so far, and the red lines the same for articles. The lines must meet at $(0,0)$ and $(1,1)$ by definition. When the blue line is above the red, the number of citations is outpacing the number of articles for authors alphabetically before this point. Since you may be reading in old timey blacke & whyte, I will also tell you that the blue line is more or less above the red line in both pictures. This means that both show a bias towards the early alphabet, with the forward direction being more dramatic (for basically the entire range). The backward range briefly shows the opposite bias (crossing over at author #5,400,000, "ojars biskaps"), suggesting that the best names may be alphabetically early but in the *B*s, not *A*s. The poorer performance of A-names may be due to authors attempting to "game the system" by publishing with names like "a a."

- Sometimes the last page of a manuscript is "lost," truncating its alphabetical bibliography to some prefix.

- Some academics try to `fread` all papers into memory, but it fails on the file that's just slightly larger than 2^{31} bytes, and so the papers therein don't get cited, and those papers are alphabetically in the middle or end of the alphabet.

- When applying for grad school, tenure track faculty jobs, grants, and so on, packets are alphabetized "for fairness," and the mood of the committee becomes more irritable as they make progress through the set.

- Decreased access to pie and exposure to unfair situations in youth cause a lifelong predisposition to failure.

5 I now bore of the titular topic. Let us discuss new topics.

Specifically, let's discuss titles. Some have suggested that the titles of papers also affect their citation rates, and it is reasonable to suspect that an alphabetical bias could apply here as well. After applying some light normalization, I extracted each of the words that appear in the titles of all papers, counting both the articles (an article is only counted once per word, even if that word appears multiple times in its title) and citations. This tells us what the expected number of citations for an article is (assuming independence, which is—as usual—completely false) when its title contains a word. Considering only words that appeared in at least 100 articles, here are the top by expectation:

word	**articles**	**citations**	**E = c/a**
folin	172	209,642	1,211.8
power/knowledge	111	55,992	499.9
{s}	101	38,241	374.9
1972-1977	171	54,917	319.2
\sqrt{s}	126	33,253	261.8
eigenfaces	176	45,284	255.8
america"s	119	30,478	253.9
gromacs	125	30,865	244.9
neo-ffi	102	21,992	213.5
esh	156	33,142	211.1
eqs	128	25,504	197.7
charmm	175	31,579	179.4
position-specific	378	67,250	177.4
kegg	201	32,989	163.3
arlequin	161	26,318	162.4
praat	158	25,217	158.6

These are mostly explained by being relatively rare words that appear in extremely popular papers. For example, *folin* comes from a 1951 paper[3] describing a procedure for using this reagent, which I guess everyone who uses this technique cites (Google scholar has this paper at 215,329 citations). *Power/knowledge* and *1972-1977* both come from the title of a Foucault book[4] with 31,590 citations. `america''s` is not a LaTeX disaster; it really has two apostrophes in it, and it is weird that this typo appears in 119 articles.

Considering just the alphabetizable words, these too show a bias towards early words (Figure 2). Bibliographies that are alphabetized by title could exhibit the same effects discussed in Section 4. Additionally, since some scholars learn English by reading the dictionary front to back, they may simply be more practiced at early words and find them more attractive or easier to understand.

It is a bit easier to interpret the expected citation numbers if we normalize them. The average citation rate of all articles in this set is 5.1261. This may seem high, especially considering the 90 million articles with no citations, but keep in mind that most papers have bibliographies in which they cite several others. Data quality issues aside, this is basically the same as saying that the averge bibliography length is 5.1261. We can assign each word a "citation multiplier" effect now, defined as simply

$$\text{multiplier}_w = \frac{\text{citations}_w/\text{articles}_w}{5.1261}$$

and now numbers greater than 1 improve your chances at citation, and numbers less than 1 diminish it. Here are the most common words:

	artices	citations	multiplier
1. of	64,322,278	366,756,237	1.207822
2. and	42,576,889	276,526,184	1.375780
3. in	39,685,539	230,534,759	1.230526
4. the	38,873,614	216,491,707	1.179704
5. a	22,666,824	139,244,700	1.301292
6. for	19,048,050	114,197,875	1.269972
7. on	15,194,275	68,904,896	0.960631
8. with	10,825,547	60,752,792	1.188784
9. to	10,202,749	58,590,695	1.216461
10. de	7,753,205	3,283,786	0.089718
11. by	7,259,598	42,234,470	1.232370
12. from	5,389,957	35,999,433	1.414806
13. an	5,368,923	33,088,995	1.305519
14. la	4,417,389	1,471,023	0.070541
15. study	4,370,395	22,022,172	1.067397

Several of these words are visible in Figure 2, in fact; they are so common that they cause large jumps in both article and citation count. Curiously, most of the common words have a multiplier greater than 1; they *increase* the expected number of citations. *Of* sounds smart (e.g. *Of Mice And Men*), *and* makes sense (paper contains at least two results). On the other hand, *on* actually reduces the citation count, probably because it implies equivocation (e.g. *On the other hand*). Dramatically, common non-English words have very low multipliers; Spanish words *de* and *la* reduce citations by a factor of more than ten. This may be due to problems in the data set, but it is certainly easy to imagine real cultural bias!

Finally, here are the words with the worst multipliers:

313779. 書評	29,582	1	0.000068
313778. インタビュー	20,547	1	0.000097
313777. facharzt	85,315	9	0.000117
313776. f3d	125,257	15	0.000128
313775. recenzja	63,568	8	0.000142
313774. newhaven	13,296	1	0.000150
313773. secrétaire	12,267	1	0.000163
313772. zahnarzt	17,958	2	0.000167
313771. libguides	389,727	72	0.000187
313770. 研究	12,961	2	0.000231
313769. ュ	15,984	3	0.000250
313768. kitas	7,713	1	0.000259

For this set, I only considered words that appeared in the title of at least one paper that was actually cited. The worst multiplier is 書評, which is Chinese for "book review;" it reduces your chances of citation by more than ten thousandfold. Close by is インタビュー, Japanese for "interview." *Facharzt* is like (medical) "specialist" in German, *recenzja* is "review" in Polish, and *newhaven* is a misspelling of a city in Connecticut

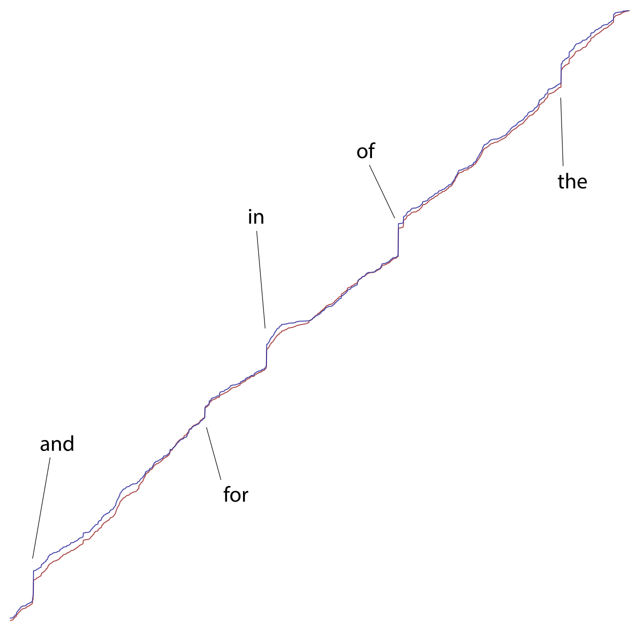

Figure 2: CDF for words arranged alphabetically. This is just as in Figure 1, but since each entry is a single word, there is no need to treat forward and backward differently. Again, we see a small alphabetical bias, with citations leading articles for most of the range. In fact, papers that include the very first word, “a”, are cited 1.3× as often as the average paper. Other common and advantageous words are visible as clear bumps.

where I grew up and was unfairly and repeatedly placed in the middle of lines. But most of these are just generic non-English words (研究 is Chinese for “research” and 그 Korean for “that”).

6 What is the least citable paper?

Thinking about first and last, one could reasonably ask: Of all papers, what are the least citable ones? Of course, the paper should have never actually been cited, but in the database there at least 90 million articles with no citations, all tied for this distinction. Can we rank among these papers to break the tie?

(Of course: In case someone who wrote one of these papers finds themselves here, please note that I am just poking fun, not seriously declaring these articles somehow “worst.” Many good articles are never cited. For example, as far as I know, the paper you are reading right now has no citations.)

One way to do this is to try to estimate the number of citations that a paper will get based on the words in its title. Since we already collected citation multipliers for all words, this is an apparently simple affair. We can estimate the citation rate by just taking the product of the multipliers for the words. We treat words for which we don’t have data (because they didn’t meet thresholds) as having multiplier 1.

This first cut has some problems. As an illustration, the article with the highest estimated citation rate (among those with zero actual citations) is a paper[5] called

> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid Grid Grid Grid Grid Grid
> Grid Grid Grid Grid

which according to the analysis is expected to have 5,262,751,498,750,000,000,000,000 citations. This is because the word *grid* has a favorable multiplier and is repeated 121 times.

So I limited it to articles with no more than 20 words in the title (arbitrary). Now the least citable articles all look like this:

> IMPLEMENTASI PERATURAN MAHKAMAH AGUNG NOMOR 1 TAHUN 2008 TENTANG PROSEDUR MEDIASI DI PENGADILAN (STUDI PUTUSAN MEDIASI DI PENGADILAN NEGERI SAMARINDA)[5]

Which appears to be a real dissertation[6] (bachelor’s degree of law) from a student in Indonesia, but there are literally thousands of similar looking papers. Given its company, it may very well be truly the least citable

[5]Translated: IMPLEMENTATION OF REGULATION OF THE SUPREME COURTS NUMBER 1 OF 2008 ABOUT MEDIATION PROCEDURES IN THE COURT (STUDY OF DECISION MEDIATION IN THE COURT SAMARINDA COUNTRY)

paper, or it may be a systematic data problem, or even spam. But I was hoping for some English language articles so that we could appreciate the joke together in the native tongue.

Even filtering for articles with `"lang"` explicitly set to `"en"`, I needed to manually create a blacklist of hundreds of non-English words (mostly Indonesian and Polish) that commonly appeared in these article titles. After a few rounds of blacklisting, some "English" articles started to appear:

> Office Hours: Monday, Wednesday, Thursday and Friday 9:00 a.m. to 5:00 p.m.; Tuesday 9:00 a.m. to 6:45 p.m
> *(citation multiplier* 5.37×10^{-27}*)*

> Thomas Aquinas: Theologian By Thomas F. O'Meara, O.P. Notre Dame, University of Notre Dame Press, 1997. 302 pp. $16.95
> *(citation multiplier* 6.05×10^{-27}*)*

The first is probably not the title of an article[7], although we can agree that it is unlikely to be cited. The second is probably a bad entry, with the entire citation packed into the title field. This book actually does have 94 citations (or DOES IT? [8]) on Google Scholar and an Amazon Sales Rank of #10,837 in *Christian Denominations & Sects*.[6] In order to prevent such data problems from affecting our results, I then added a requirement that the article have basic metadata (either a Digital DOI Identifier field or `venue`) for it to be considered. There vast majority still appear to be broken entries, but picking through the results I'm finally able to find what appear to be actual articles:

> Eight contemporary poets : Charles Tomlinson, Donald Davie, R.S. Thomas, Philip Larkin, Ted Hughes, Thomas Kinsella, Stevie Smith, W.S. Graham
> *(citation multiplier* 5.81×10^{-26}*)*

> The 80th Birthday of Sir Henry Dale, O.M., G.B.E., M.D., F.R.C.P., F.R.S.: Salute to Henry Hallett Dale
> *(citation multiplier* 1.75×10^{-19}*)*

> Narratives Unsettled: Digression in Robert Walser, Thomas Bernhard, and Adalbert Stifter by Samuel Frederick (review)
> *(citation multiplier* 2.74×10^{-19}*)*

[6]On the other hand, the entry does have a different author[9], so it could possibly be a review of this book whose title is literally the string above, including the price?

> Catherine Mowry LaCugna's Trinitarian Theology: III- The Ecumenical Implications of Catherine Mowry Lacugna's Trinitarian Theology *(citation multiplier* 5.06×10^{-19}*)*

> Texts Illustrating the Causality of the Sacraments from William of Melitona, Assisi Bibl. Comm. 182, and Brussels Bibl. Royale 1542
> *(citation multiplier* 1.70×10^{-18}*)*

> Shakespeare's Titus Andronicus and Bandello's Novelle as Sources for the Munera Episode in Spenser's Faerie Queene, Book 5, Canto 2 *(citation multiplier* 3.41×10^{-18}*)*

I don't know about you but I fell asleep just reading the titles. The first is definitely a real article [10]—which apparently has 95 citations on Google Scholar—so the estimate failed us here. It's easy to see why it has such a bad score, though; it's packed with people's names and initials, just like the scores of uncited book reviews and other broken citations. The second is also a real tribute to this Nobel prize winner [11] with 5 citations that Google knows of. Again it contains names and appelations that look like initials.

The third is a book review [12] of a book [13] with itself only 11 citations (makes sense, as the title contains three people's names) and indeed has no citations that Google knows of, so it is a solid contender. After that some theological stuff [14, 15] each with 3 citations on Google, and another literary analysis [16] which has 2.

Lessons learned:

- Taking a noisy database of tens of millions of entries and then trying to dig through the bottom of the barrel for something "funny" or even "interesting" is tough going. Maybe this should have been obvious.
- The least-cited works take on some recurring forms:
  - Memorials for dead or old guys with lots of honorifics
  - Articles with people's names in the title
  - Reviews of books, or books about other literary work, or reviews of books about other literary work
  - Articles about God stuff or God People
  - Articles not in English
  - Broken citations
  - Combinations of the above.

Enterprising academics would do well to avoid such areas of research, in addition to using alphabetically early words and author names in their publications.

- As one downside, this approach does not really work, given that several of these articles have a few actual citations upon comparing to another database.

Of course, since these papers are all notable for being so extremely uncitable, the true least citable papers are the **next least citable**, according to this analysis. To avoid paradoxes, the reader is discouraged from trying to reproduce these results.

6.1 Alliterative Articles

Adopting an all-'A' affect also advances articles. Although affairs are abbreviated, adequate abstracts abound: Astronomy, asthma, atmospheric acclimatization, autobiography, Azerbaijani accents, acceleration, and abundant alternatives. After achieving academic accomplishments, acknowledge A.A.'s *Academic Advancement Advice: Author Articles as A. A*!

7 Bibliography

References

[1] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *KDD'08*, 2008, pp. 990–998.

[2] M. Yip, "Native JSON benchmark," 2018, https://github.com/miloyip/nativejson-benchmark.

[3] O. H. Lowry, N. J. Rosebrough, A. L. Farr, and R. J. Randall, "Protein measurement with the Folin phenol reagent," *Journal of biological chemistry*, vol. 193, no. 1, pp. 265–275, 1951.

[4] M. Foucault, *Power/knowledge: Selected interviews and other writings, 1972–1977*. Pantheon, 1980.

[5] J. Leigland and H. Russell, "Grid grid," 2009.

[6] A. A. PUTRI, "IMPLEMENTASI PERATURAN MAHKAMAH AGUNG NOMOR 1 TAHUN 2008 TENTANG PROSEDUR MEDIASI DI PENGADILAN (STUDI KASUS DI PENGADILAN NEGERI BOYOLALI)," Sarjana S1 dalam Ilmu Hukum, UNIVERSITAS SEBELAS MARET SURAKARTA, 2009.

[7] M. D. J. Johnson, J. Coyle, R. Kelsey, J. Fason, D. O. F. Omole, J. Herbert-Carter, K. Carter-Wicker, T. Jones, D. Mack, B. Malvea, Y.-X. Meng, M. Nichols, B. Taylor, O. Okuwobi, G. Strayhorn, H. Strothers, M. Smith, and M. D. MMA, "Office hours: Monday, wednesday, thursday and friday 9:00 a.m. to 5:00 p.m.; tuesday 9:00 a.m. to 6:45 p.m."

[8] T. F. O'Meara, *Thomas Aquinas, Theologian*. University of Notre Dame Press, 1997.

[9] w b stevenson, "Thomas Aquinas: Theologian By Thomas F. O'Meara, O.P. Notre Dame, University of Notre Dame Press, 1997. 302 pp. $16.95," *Theology Today*, 2000.

[10] C. Bedient, *Eight Contemporary Poets: Charles Tomlinson, Donald Davie, RS Thomas, Philip Larkin, Ted Hughes, Thomas Kinsella, Stevie Smith, WS Graham*. London; New York [etc.]: Oxford University Press, 1974, vol. 358.

[11] O. Loewi, "The 80th birthday of sir henry dale, om, gbe, md, frcp, frs: Salute to henry hallett dale," *British Medical Journal*, vol. 1, no. 4926, p. 1356, 1955.

[12] T. Holmes, "Narratives unsettled: Digression in robert walser, thomas bernhard, and adalbert stifter by samuel frederick," *Monatshefte*, vol. 107, no. 2, pp. 330–332, 2015.

[13] S. Frederick, *Narratives Unsettled: Digression in Robert Walser, Thomas Bernhard, and Adalbert Stifter*. Northwestern University Press, 2012.

[14] A. J. Torrance, "Catherine mowry lacugna's trinitarian theology: Iii-the ecumenical implications of

catherine mowry lacugna's trinitarian theology," *Horizons*, vol. 27, no. 2, pp. 347–353, 2000.

[15] K. F. Lynch, "Texts illustrating the causality of the sacraments from william of melitona, assisi bibl. comm. 182, and brussels bibl. royale 1542," *Franciscan Studies*, vol. 17, no. 2/3, pp. 238–272, 1957.

[16] J. Fitzpatrick, "Shakespeare's Titus Andronicus and Bandello's Novelle as Sources for the Munera Episode in Spenser's Faerie Queene, Book 5, Canto 2," *Notes and Queries*, vol. 52, pp. 196–198, 2005. [Online]. Available: http://dx.doi.org/10.1093/notesj/gji221