Popularity Ranking for Scientific Literature Using the Characteristic Scores and Scale Method

Philipp Schaer[1] and Narges Tavakolpoursaleh[2]

TH Köln (University of Applied Sciences), Cologne, Germany
philipp.schaer@th-koeln.de
GESIS – Leibniz Institute for the Social Sciences, Cologne, Germany
narges.tavakolpoursaleh@gesis.org

1 Introduction

The TREC 2016 OpenSearch track is focused on ad-hoc search for scientific literature. Three scientific search engines and document repositories were part of this living lab-centered evaluation campaign: (1) CiteSeerX, (2) Microsoft Academic Search, and (3) SSOAR - Social Science Open Access Repository. The authors of this paper are also responsible for the implementation of the living lab infrastructure and the LL4IR API that is necessary to include an online system into the OpenSearch evaluation campaign. This work is based on a Master's thesis at University of Bonn [7]. Implementation details can be found there and in the lab's overview paper [1] and from a higher perspective in [6].

In this paper we will present our work on popularity-based relevance ranking within the two systems CiteSeerX and SSOAR. Both offer different types of usage and popularity data. We would like to test a normalization method for these kind of data known as the Characteristic Scores and Scale Method (CSS).

2 Method

In our first Living Lab paper from 2015 [5] we used historical usage data (click-through rates of the online toy store Regio Játék) to augment a Solr-based relevance ranking of the available products in the online catalogue. We simply used the log of the raw click-through rates as a boosting factor with-in Solr's ranking formula. This straight-forward application of usage and popularity data is considered being to simplistic as it introduces some flaws and drawbacks, like:

– the biases raw data introduces into the ranking due to e.g. different publication dates;
– the lack of a common scale to compare the different usage data with each other.

Biases are an immanent problem of usage data. Take the example of the raw click-through rate for a given product in the online toy store catalogue. A product that

Table 1: Overview on different types of usage data found in systems and evaluation campaigns

System	Available usage data	Used in
Regio Játék	Historical click-through data	CLEF 2015 LL4IR [5]
SSOAR	Number of document views and PDF full text downloads	TREC 2016 OpenSearch
CiteSeerX	Number of citations	TREC 2016 OpenSearch

is quite new is not able to gather as many clicks as a best-selling item that is part of the catalogue for quite some time. This is directly connected to the second issue: A direct comparison of the click-through rates without a normalization or homogenization is not possible.

Click-through data is not the only way to measure usage. In table 1 we see the three systems used in last year's CLEF LL4IR workshop and this year's TREC OpenSearch workshop. All systems offer a different set of usage data: Click-through data in Regio Játék, document views and PDF download rates in SSOAR, and citation counts in CiteSeerX.

We therefore need a method to normalize the usage data to remove biases and to enable some kind of comparability. In the next section we will present a simple, yet effective way to normalize the usage data distributions.

2.1 Normalizing Usage Data Distributions with the Characteristic Scores and Scale Method

Our method is based on a procedure called the Characteristic Scores and Scales method (CSS) described by Glänzel [2]. The CSS method is used to find characteristic partitions for citation distributions. They used the method to establish classes of papers that they interpreted as "poorly cited", "fairly cited", "remarkably cited", or "outstandingly cited". These partitions can then be used to normalize different kinds of usage data distributions.

As described by Plassmeier et al. [3] these classes are constructed by calculating the class boundaries in the following way: First we take the mean of the distribution $\beta_1 = \mu$. The distribution is truncated at the first boundary and the second boundary is found at the mean of the truncated distribution, $\beta_2 = \text{mean}(x_i | x_i \leq \beta_1)$. For the rest of the distribution the next k-th class boundary is defined by

$$\beta_k = \text{mean}(x_i | x_i \leq \beta_{k-1}). \tag{1}$$

This method iterates as long as a previously defined threshold of number of classes has been reached or if the number of elements in class k drops below a previously defined threshold. The found classes are used to construct a continuous transformation function of the original distribution values to the normalized values. To do so, we map the class boundaries to the interval $[0, 1]$ and linearly interpolate between the class boundaries, i.e. $\beta_k^{'} = k/k_{max}$.

Plassmeier et al. [3] showed that this normalization steps can be used for different kinds of usage data distributions, like number of citations, number of record views or the number of loans at local libraries. We therefore were eager to learn if we can apply them to the usage data found in SSOAR or CiteSeerX.

2.2 Popularity-based Relevance Ranking for SSOAR and CiteSeerX

As listed in table 1 we extracted three different sets of usage data: Document views and PDF download rates from SSOAR and citation counts from CiteSeerX. We gathered these data by using internal APIs of SSOAR[1] and by using web scraping technologies in the case of CiteSeerX. The usage data dates to the middle of May 2016. The data was gathered once and was not updated during the campaign. In table 2 we see an overview on the popularity data used in the following section.

We define our new popularity-based relevance rank scoring $\mathrm{s}_{spop}(q, d, U_d)$ for a given query q, a specific document d and the set of popularity and usage values U_d for that given document. U_d can contain different values like u_{down} being the download, u_{view} being the view counts for each document, or u_{cite} being the citation count for this document.

$$\mathrm{s}_{spop}(q, d, U_d) = \mathrm{s}_{solr}(q, d) * \mathrm{s}_{pop}(d, U_d) \tag{2}$$

The original Solr score[2] is augmented with a document dependent popularity score $\mathrm{s}_{pop}(d, U_d)$ that is defined a follows:

$$\mathrm{s}_{pop}(d, U_d) = (\sum_{i \in U_d} s_{css_i}(d, u_{d,i})) + 1. \tag{3}$$

For SSOAR and it's two popularity values this is:

$$\mathrm{s}_{pop}(d, U_d) = (\mathrm{s}_{css}(d, u_{d,view}) + \mathrm{s}_{css}(d, u_{d,down})) + 1. \tag{4}$$

For CiteSeerX and it's citation counts this is:

$$\mathrm{s}_{pop}(d, U_d) = \mathrm{s}_{css}(d, u_{d,cite}) + 1. \tag{5}$$

The popularity values $\mathrm{s}_{css}(d, u_{d,view})$, $\mathrm{s}_{css}(d, u_{d,down})$, and $\mathrm{s}_{css}(d, u_{d,cite})$ which are dependent on the document view, download and citation rate are calculated on basis of the CSS method described earlier in section 2.1 and are summed up. In case that there is no usage data available the value of $\mathrm{s}_{pop}(d, U_d)$ is 1. Therefore with no usage data available $\mathrm{s}_{spop}(q, d, u_d, u_v) = \mathrm{s}_{solr}(q, d)$, or in other word the original Solr ranking score is not modified.

[1] As we had direct access to the SSOAR productive system, we were able to get these data without using any additional steps like web scraping.

[2] See implementation details at `https://lucene.apache.org/core/2_9_4/api/core/org/apache/lucene/search/Similarity.html`

Table 2: Corpus statistics on the popularity data gathered from SSOAR and CiteSeerX.

	SSOAR downloads	SSOAR views	CiteSeerX cites
# docs total	24,760	24,760	9,724
# docs w. usage data	21,523	24,724	4,682
max	504,720	21,788	11,648
sum	6,549,674	9,822,049	3,545,420
avg	264.505	396.658	364.605

The behaviour of our boosting method is verbalized as follows: A high download rate in combination with a high document view rate leads to a general high boost rate of s$_{pop}$ while a single high value of one of these alone is not enough to guarantee a significant boost and without any usage data available the original Solr relevance ranking is not altered.

In our experiments we used three different popularity values: record views, PDF downloads and citation counts. In a literature-related field like TREC OpenSearch other popularity values are available, as listed by [3]:

- Citation related values, like number of citation of an item or the citation impact for a journal or publication venue;
- Author metrics, like h-index or other citation-reltated impact mesuares for authors;
- Usage data, like number of record views, number of clicks on full text or downloads, or number of library loans.

3 Results

In table 2 we can see the results of our popularity data crawling for SSOAR and CiteSeerX. For SSOAR we analyzed approx. 25,000 documents and 10,000 for CiteSeerX. On average there were 264 downloads and 396 record views per document for SSOAR and 364 citations for the CiteSeerX documents. Other corpus statistics are listed in the table.

3.1 Impact on the Usage Date Distributions

We applied the CSS normalization method on the usage data of SSOAR. Due to an error in our submission process we uploaded a wrong CiteSeerX ranking. This wrong submission used raw citation counts not the CSS normalized values. Therefore we can not directly compare the values in the following section. In the two figures 1 and 2 we can see the effects of this normalization method for SSOAR data. The highly skewed and scattered data distributions are smoothed and values get more comparable. In the figure we see the usage data of five different years of publication: 2005, 2007, 2009, 2011 and 2013. On the double-logarithmic scale we see that the raw download and view numbers for the different publication year's differ. The range of the raw numbers is mostly between 10^3 and 10^4. After applying the CSS normalization these values drop between 10^2 and 10^3 while having generally fewer differences between the years. We can see that the biases introduced by the different publication years are not as present as in the raw data distributions. Nevertheless the distributions keep their skewed characteristics which is usually a good thing if we want to use these data for re-ranking purposes [4].

3.2 Impact on the Document Ranking

To quantify the impact of our adjusted ranking formula we calculated Kendall's τ comparing the rank correlation between the original ranking of the two systems SSOAR and CiteSeerX and our experimental ranking. For SSOAR the original ranking is a more or less standard Solr ranking without any special modifications. For CiteSeerX the original ranking is that of the original platform. Our experimental ranking is the CSS-normalized popularity ranking described in section 2.2 using full text downloads and record view counts for SSOAR and raw citation rates for CiteSeerX.

For SSOAR the two different rankings are totally different and more-or-less independent from each other with a τ-value of -0.013. The minimum τ-value of -1 is measured for query `ssoar-q62` and the maximum τ-value of 0.666 for query `ssoar-q42`. The absolute distribution of τ-values can be seen in figure 3. With most of the values being in the range of $[-0.2, 0.2]$ we see that most of the rankings are different, showing the very high impact of our new weighting method.

For CiteSeerX the impact is quantified with an average τ-value of 0.491. The minimum τ-value of 0.021 is measured for query `citeseerx-q137` while the maximum τ-value of 1 is recored for queries `citeseerx-q190` and `citeseerx-q127`. In contrast to the clear and obvious re-ranking behaviour in the SSOAR use case the influence on the CiteSeerX rankings is not that big. Please keep in mind that these values are based on a wrong submission. They are therefore not suitable for direct comparison with the rankings in SSOAR.

4 Conclusion

We could see a high re-ranking impact on the SSOAR ranking using our CSS-normalized popularity ranking method. In CiteSeerX this impact is also measurable but not as clear as in the SSOAR use case. This can be explained with the fact that in SSOAR there is just a plain Solr text-based relevance ranking while CiteSeerX itself uses popularity measures (citation counts) to influence their default ranking and of course our submission process error as mentioned before. Nevertheless the high number on average citation per CiteSeerX document (364 citations) is an indicator for the citation-based ranking method used in CiteSeerX. The candidate documents are generally highly-cited documents,

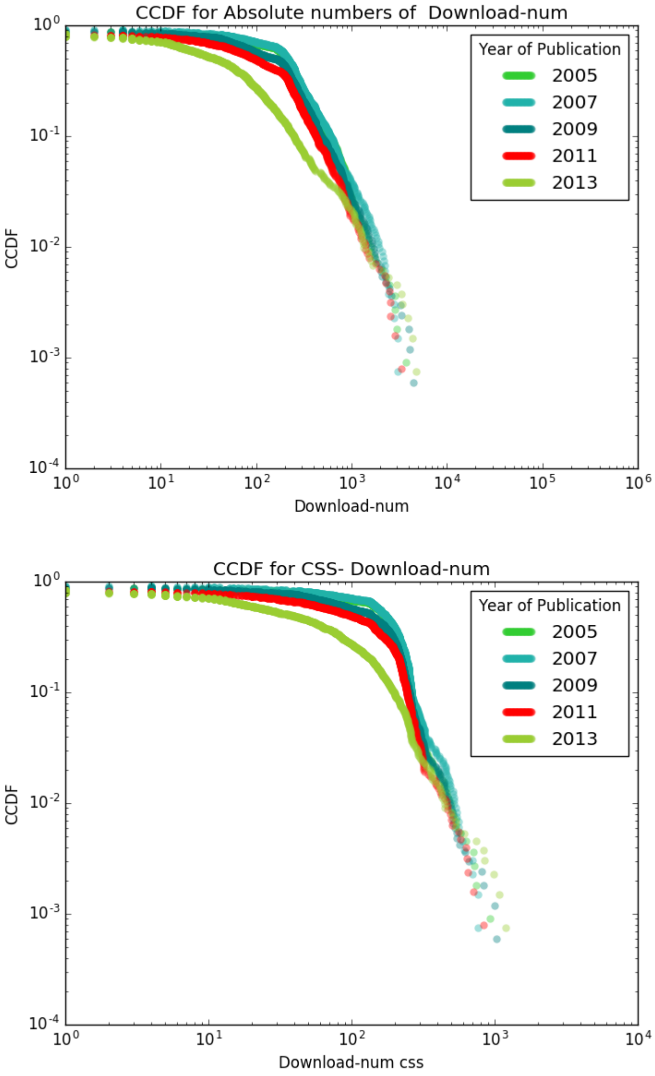

Fig. 1: Plots of the counter cumulative distribution function of *document downloads* for three years of publication 2007, 2009, and 2014. Top: Absolute number of document downloads. Bottom: Smoothed numbers using the CSS method.

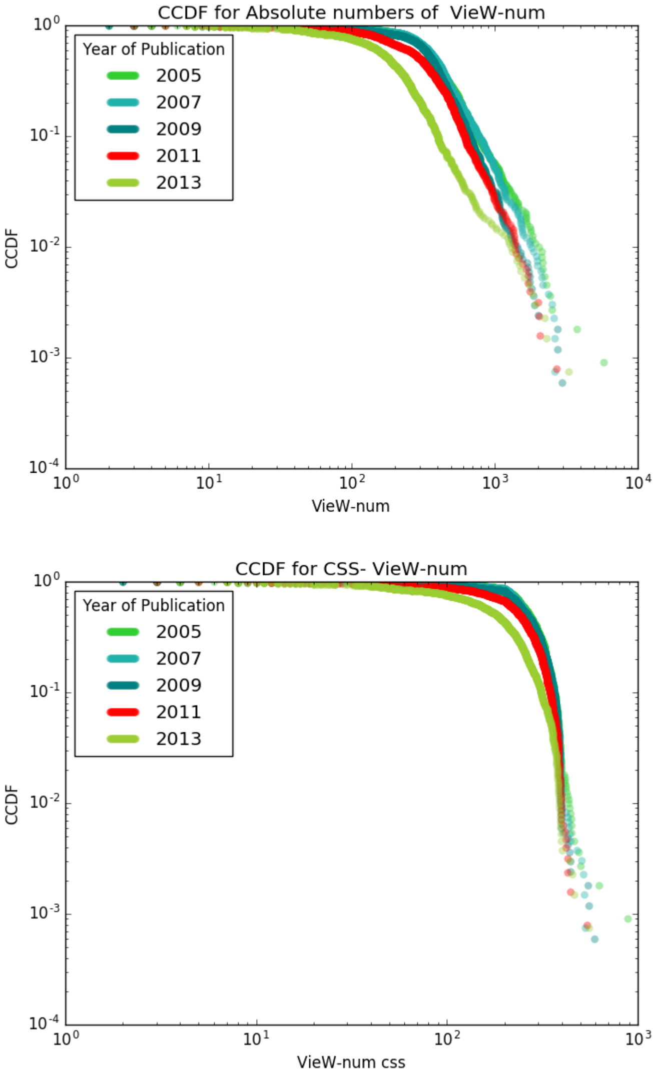

Fig. 2: Plots of the counter cumulative distribution function of *document views* for three years of publication 2007, 2009, and 2014. Top: Absolute number of document views. Bottom: Smoothed numbers using the CSS method.

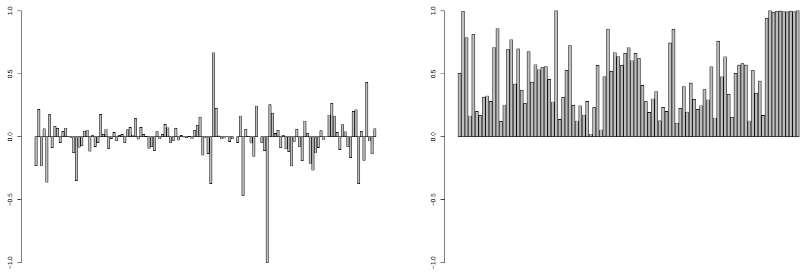

Fig. 3: Plot of Kendall's τ values for each query from round #1 and #2 for SSOAR (left) and CiteSeerX (right).

making our popularity-ranking method not as effective compared to the simple text-based ranking of SSOAR where usage data and the popularity ranking idea can show some effects.

In fact the CSS normalized popularity ranking approach performed best in TREC OpenSearch 2016. However we must admit that the number of wins was to low to generalize on this single living lab evaluation.

5 Acknowledgement

We would like to thank the whole SSOAR team that supported us to implement the LL4IR API into the productive system and opening up the system for academic research. They were patient and big-hearted during the deployment phase when some bugs seriously hit the server. Thank you for your understanding and your support.

This work was supported by the ESF Research Networking Programme ELIAS.

References

1. Balog, K., Schuth, A., Tavakolpoursaleh, N., Schaer, P., Chuang, P.Y., Wu, J., Giles, C.L.: Overview of the trec 2016 open search track. In: Proceedings of the Twenty-Fifth Text REtrieval Conference (TREC 2016). NIST (2016)
2. Glänzel, W.: Characteristic scores and scales: A bibliometric analysis of subject characteristics based on long-term citation observation. Journal of Informetrics 1(1), 92–102 (Jan 2007)
3. Plassmeier, K., Borst, T., Behnert, C., Lewandowski, D.: Evaluating popularity data for relevance ranking in library information systems. In: Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community. p. 125. American Society for Information Science (2015), `http://dl.acm.org/citation.cfm?id=2857195`

4. Schaer, P.: Der Nutzen informetrischer Analysen und nicht-textueller Dokumentattribute für das Information Retrieval in digitalen Bibliotheken. Ph.D. thesis, Universität Koblenz-Landau (May 2013), http://kola.opus.hbz-nrw.de/frontdoor.php?source_opus=896&la=de
5. Schaer, P., Tavakolpoursaleh, N.: Historical clicks for product search: Gesis at clef ll4ir 2015. In: Cappellato, L., Ferro, N., Jones, G.J.F., SanJuan, E. (eds.) Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015. CEUR Workshop Proceedings, vol. 1391. CEUR-WS.org (2015), http://ceur-ws.org/Vol-1391/26-CR.pdf
6. Schaer, P., Tavakolpoursaleh, N.: Ideas for a standard ll4ir extension - living labs from a system operator's perspective. In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (eds.) Working Notes of CLEF 2016 - Conference and Labs of the Evaluation forum, Évora, Portugal, 5-8 September, 2016. CEUR Workshop Proceedings, vol. 1609, pp. 591–592. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1609/16090591.pdf
7. Tavakolpoursaleh, N.: A Living Lab Evaluation Envirtonment for Academic Document Repositories. Master's thesis, University of Bonn, Germany (2016)