Apache Lucene as Content-Based-Filtering Recommender System: 3 Lessons Learned

Stefan Langer[1] and Joeran Beel[2,3]

[1]Otto-von-Guericke University, Department of Computer Science, Magdeburg, Germany
`langer@ovgu.de`
[2]Trinity College Dublin, Department of Computer Science, ADAPT Centre, Ireland
`joeran.beel@adaptcentre.ie`
[3]National Institute of Informatics, Digital Content and Media Sciences Research Division, Tokyo, Japan
`beel@nii.ac.jp`

Abstract. For the past few years, we used Apache Lucene as recommendation framework in our scholarly-literature recommender system of the reference-management software Docear. In this paper, we share three lessons learned from our work with Lucene. First, recommendations with relevance scores below 0.025 tend to have significantly lower click-through rates than recommendations with relevance scores above 0.025. Second, by picking ten recommendations randomly from Lucene's top50 search results, click-through rate decreased by 15%, compared to recommending the top10 results. Third, the number of returned search results tend to predict how high click-through rates will be: when Lucene returns less than 1,000 search results, click-through rates tend to be around half as high as if 1,000+ results are returned.

Keywords: recommender systems, apache lucene, content-based filtering, lessons learned

1   Introduction

Apache Lucene/Solr is probably the most common search framework, and it is frequently used by content-based-filtering recommender systems (Bancu et al., 2012; Caragea et al., 2014; Garcia Esparza, O'Mahony, & Smyth, 2010; Jonnalagedda, Gauch, Labille, & Alfarhood, 2016; Livne, Gokuladas, Teevan, Dumais, & Adar, 2014; Mitzig et al., 2016; Phelan, McCarthy, & Smyth, 2009; Pohl, 2007; Pursel et al., 2016; Shelton, Duffin, Wang, & Ball, 2010). Lucene's build-in recommendation method, which uses a classic TF-IDF-weighted term-vector retrieval approach, is also generally used frequently as a baseline method that typically achieves good results. (Demner-Fushman et al., 2011; Gipp, Meuschke, & Lipinski, 2015; Schwarzer et al., 2016)

We used Lucene to implement a research-paper recommender system in *Docear* (Beel, Gipp, Langer, & Genzmehr, 2011; Beel, Gipp, & Mueller, 2009; Beel, Langer, Genzmehr, & Nürnberger, 2013; Beel, Langer, Gipp, & Nürnberger, 2014). Docear is a free and open-source reference manager, comparable to tools like Endote, Zotero,

Mendeley, or Citavi. Docear has approximately 50,000 registered users and uses mind-maps to manage PDFs and references. Since 2012, Docear has been offering a recommender system for 1.8 million publicly available research papers on the web. Recommendations are displayed as a list of ten research papers, showing the title of the recommended papers (**Fig. 1**). Clicking a recommendation opens the paper's full-text (PDF) in the user's web browser. Between 2012 and 2015, the recommender system delivered around one million recommendations to more than 7,000 researchers. For more details on the recommender system please refer to Beel et al. (2014).
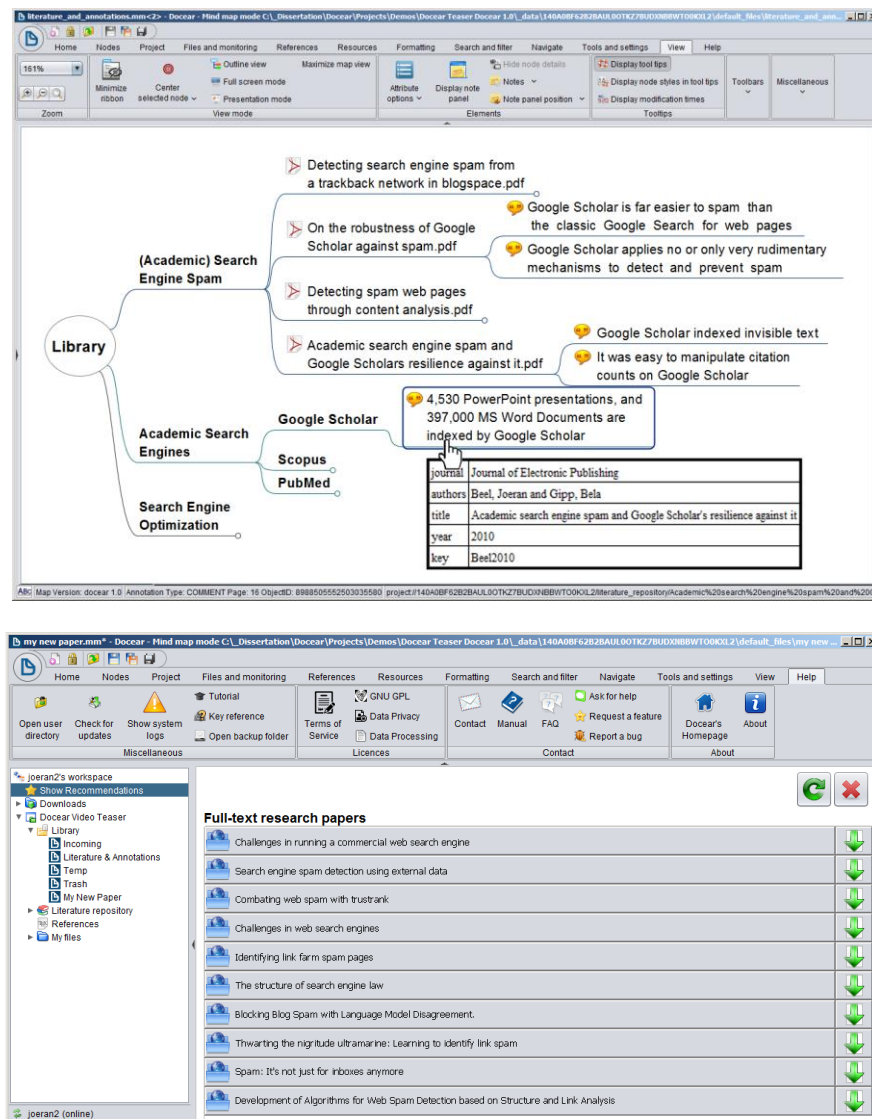

Fig. 1. Screenshot of Docear and the recommender system

In this paper, we share some experiences we made with Lucene, focusing on three aspects. First, we analyze the meaning of Lucene's relevance scores. Second, we analyze how effective recommendations are based on Lucene's suggested rank. Finally, we analyze the relationship between the amount of recommendation candidates that Lucene returns and the recommendation effectiveness. Although we did our research in the context of research-paper recommendations, results might also be interesting for other recommender-systems domains that use Lucene, for instance, in the domains of news recommender systems, website recommender systems, or tweet recommender systems (Chen, Ororbia, Alexander, & Giles, 2015; Duma, Liakata, Clare, Ravenscroft, & Klein, 2016; Garcia Esparza et al., 2010; Jonnalagedda et al., 2016; Mitzig et al., 2016; Phelan et al., 2009; Shelton et al., 2010).

2 Methodology

All presented results are based on data that we collected between May 2013, and October 2014. During this time, Docear's recommender system delivered 418,308 recommendations to 4,674 unique users. We use click-through rate as measure for the effectiveness of delivered recommendations. Click-through rate (CTR) describes the ratio of clicked and delivered recommendations. For more details on click-through rate and its suitability as evaluation metric please refer to Beel & Langer (2015). All reported differences are statistically significant ($p < 0.05$) based on a two-tailed t-test.

3 Results & Discussion

3.1 Lucene's Relevance Scores

Lucene provides relevance scores for each recommendation. This information could be used, theoretically, to recommend only documents with a relevance score above a certain threshold. However, on the Web it is often reported that these scores cannot be used to compare relevancies of recommendations between different queries, or to conclude from the relevance score how relevant the search result or recommendation is overall.[1] Our data shows a slightly different picture.

In our data, the highest relevance score for a recommendation was 19.01, median was 0.16 and mean was 0.22. **Fig. 2** shows that CTR was lowest (3.36%) for recommendations with a relevance score below 0.01, and highest (6.16%) for relevance scores of 1 and above. For recommendations with relevance scores between 0.1 and 0.8, CTR remained mostly stable around 5%. Overall, there is a notable trend: CTR increases, the higher Lucene relevance scores become.

Our observation contradicts the common claims that Lucene's relevance score cannot be used to estimate a search result's absolute relevance. If, for instance, an operator of a recommender system decided that a click-through rate of at least 4% was desirable, then recommendations with a relevance score below 0.25 should probably be discarded.

[1] https://wiki.apache.org/lucene-java/ScoresAsPercentages

Similarly, our result might lead to the conclusion to only recommend documents above a certain relevance threshold, e.g. 1. However, recommending only documents with a relevance score of 1 and above is probably not sensible as only a small fraction of recommendations had a relevance score of 1 and above (0.60%). Similarly, it might seem sensible to not recommend documents with relevance scores below 0.025 as these documents had very low CTRs. However, only a small fraction of recommendations (4.27%) had relevance scores below 0.025, so this decision would barely affect the overall click-through rate.
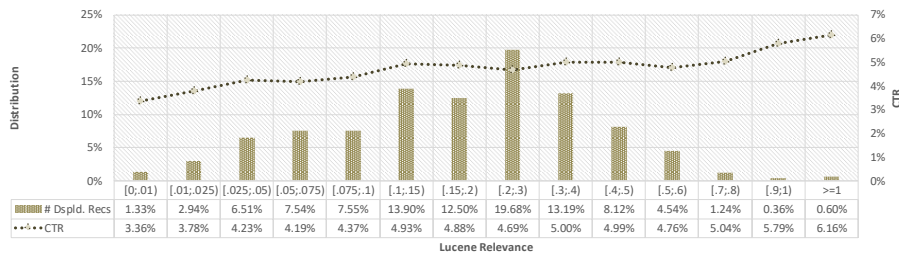
	[0;.01)	[.01;.025)	[.025;.05)	[.05;.075)	[.075;.1)	[.1;.15)	[.15;.2)	[.2;.3)	[.3;.4)	[.4;.5)	[.5;.6)	[.7;.8)	[.9;1)	>=1
# Dspld. Recs	1.33%	2.94%	6.51%	7.54%	7.55%	13.90%	12.50%	19.68%	13.19%	8.12%	4.54%	1.24%	0.36%	0.60%
CTR	3.36%	3.78%	4.23%	4.19%	4.37%	4.93%	4.88%	4.69%	5.00%	4.99%	4.76%	5.04%	5.79%	6.16%

Fig. 2. Lucene relevance score and corresponding CTR

3.2 Lucene's Rank

To increase the diversity of recommendations, Docear's recommender system randomly chose 10 recommendations out of the top50 results returned by Lucene. However, this leads to lower click-through rates. Recommendations originally being ranked 1 by Lucene received CTRs of 6.83% on average and recommendations on rank 2 received CTRs of 6.08% on average (**Fig. 3**). For ranks 3 to 10, CTR remains stable around 5.3% and then CTR constantly decreases the lower the original rank.

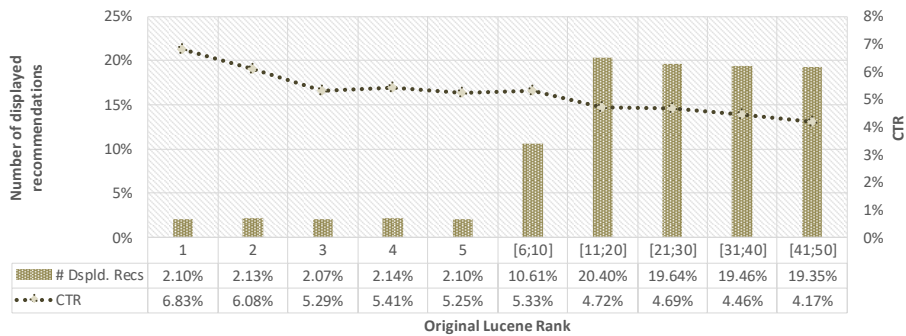|	1	2	3	4	5	[6;10]	[11;20]	[21;30]	[31;40]	[41;50]
# Dspld. Recs	2.10%	2.13%	2.07%	2.14%	2.10%	10.61%	20.40%	19.64%	19.46%	19.35%
CTR	6.83%	6.08%	5.29%	5.41%	5.25%	5.33%	4.72%	4.69%	4.46%	4.17%

Fig. 3. Lucene's rank and corresponding CTR

Overall, recommendations being in Lucene's top10 results, achieved CTRs of 5.55% on average, while the top50 achieved CTRs of 4.73% on average. This means, selecting randomly 10 recommendations from the top50 candidates decreases recommendation effectiveness by around 15%, compared to showing recommendations from the top10 only. The recommender system shuffled recommendations before they were displayed. This means, position bias cannot have influenced the results (Craswell, Zoeter, Taylor, & Ramsey, 2008; Hofmann, Schuth, Bellogin, & Rijke, 2014; Pan et al., 2007; Wang, Bendersky, Metzler, & Najork, 2016).

3.3 Number of Recommendation Candidates

By default, Lucene returns 1,000 recommendations, i.e. search results. In our data, Lucene returned the maximum possible amount of 1,000 results for 91.25% of all term-based recommendations (**Fig. 4**). In contrast, only for 0.05% of citation-based searches 1,000 results were returned. Most citation-based searches returned between one and nine results (34.84%) or between 10 and 24 results (29.94%). Click-through rates seem to be rather high when only few results were returned. For term-based searches, results are the opposite: the more recommendation candidates are available, the higher the CTR tends to be. Consequently, for term-based recommendations, the number of results might be a good approximation of recommendation effectiveness. If less than 1,000 results are returned it might make sense to no recommend the documents or try an alternative recommendation approach.
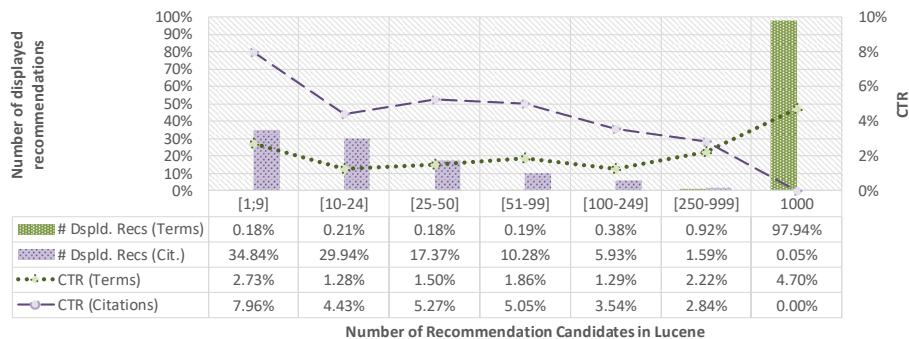
	[1;9]	[10-24]	[25-50]	[51-99]	[100-249]	[250-999]	1000
# Dspld. Recs (Terms)	0.18%	0.21%	0.18%	0.19%	0.38%	0.92%	97.94%
# Dspld. Recs (Cit.)	34.84%	29.94%	17.37%	10.28%	5.93%	1.59%	0.05%
CTR (Terms)	2.73%	1.28%	1.50%	1.86%	1.29%	2.22%	4.70%
CTR (Citations)	7.96%	4.43%	5.27%	5.05%	3.54%	2.84%	0.00%

Number of Recommendation Candidates in Lucene

Fig. 4. CTR based on the number of recommendation candidates in Lucene

4 Summary & Future Work

From our analysis, we learned three lessons. First, Lucene's relevance score allows to predict – to some extend – how relevant a recommendation will be for a user. For instance, in our scenario, it seems sensible to not recommend documents with a relevance score below 0.025. However, since only few recommendations had such a low rele-

vance score, discarding them will probably not notably affect the overall recommendation effectiveness. Second, recommending ten recommendations out of the top50 results might be sensible. Although this process decreases the overall recommendation effectiveness by 15%, the recommendation diversity or number of total recommendations is increased. Third, the number of recommendation candidates returned by Lucene is suitable to approximate the recommendation effectiveness. If Lucene returns less than 1,000 results for term-based recommendations, the click-through rate probably will be around half as high as if 1,000 candidates are returned. In the case of less than 1,000 results, it might make sense to not display the recommendations or generate recommendations again with another recommendation approach.

For the future, we suggest to repeat our analyses in different scenarios, for instance, with news recommenders or other literature recommender systems, to see if Lucene behaves in the same way as in the scenario of Docear. Currently, we are developing a recommender system as-a-service that would allow us to conduct such analyses with different partners (Beel & Gipp, 2017; Beel, Gipp, Langer, Genzmehr, et al., 2011).

5   Acknowledgements

References

Bancu, C., Dagadita, M., Dascalu, M., Dobre, C., Trausan-Matu, S., & Florea, A. M. (2012). ARSYS-Article Recommender System. *Proceedings of the 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (pp. 349–355).

Beel, J., & Gipp, B. (2017). Mr. DLib: Recommendations-as-a-Service (RaaS) for Academia (Pre-print). *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*.

Beel, J., Gipp, B., Langer, S., & Genzmehr, M. (2011). Docear: An Academic Literature Suite for Searching, Organizing and Creating Academic Literature. *Proceedings of the 11th Annual International ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, JCDL '11 (pp. 465–466). ACM. doi:10.1145/1998076.1998188

Beel, J., Gipp, B., Langer, S., Genzmehr, M., Wilde, E., Nürnberger, A., & Pitman, J. (2011). Introducing Mr. DLib, a Machine-readable Digital Library. *Proceedings of the 11th ACM/IEEE Joint Conference on Digital Libraries (JCDL '11)* (pp. 463–464). ACM. doi:10.1145/1998076.1998187

Beel, J., Gipp, B., & Mueller, C. (2009). SciPlore MindMapping' - A Tool for Creating Mind Maps Combined with PDF and Reference Management. *D-Lib Magazine*, *15*(11). doi:10.1045/november2009-inbrief

Beel, J., & Langer, S. (2015). A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. In S. Kapidakis, C. Mazurek, & M. Werla (Eds.), *Proceedings of the 19th International Conference on Theory and Practice of Digital Libraries (TPDL)*, Lecture Notes in Computer Science (Vol. 9316, pp. 153–168). doi:10.1007/978-3-319-24592-8_12

Beel, J., Langer, S., Genzmehr, M., & Nürnberger, A. (2013). Introducing Docear's Research Paper Recommender System. *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'13)* (pp. 459–460). ACM. doi:10.1145/2467696.2467786

Beel, J., Langer, S., Gipp, B., & Nürnberger, A. (2014). The Architecture and Datasets of Docear's Research Paper Recommender System. *D-Lib Magazine*, *20*(11/12). doi:10.1045/november14-beel

Caragea, C., Wu, J., Ciobanu, A., Williams, K., Fernández-Ramírez, J., Chen, H.-H., Wu, Z., et al. (2014). CiteseerX: A scholarly big dataset. *European Conference on Information Retrieval* (pp. 311–322). Springer.

Chen, H.-H., Ororbia, I., Alexander, G., & Giles, C. L. (2015). ExpertSeer: a Keyphrase Based Expert Recommender for Digital Libraries. *arXiv preprint arXiv:1511.02058*.

Craswell, N., Zoeter, O., Taylor, M., & Ramsey, B. (2008). An experimental comparison of click position-bias models. *Proceedings of the 2008 International Conference on Web Search and Data Mining* (pp. 87–94). ACM.

Demner-Fushman, D., Abhyankar, S., Jimeno-Yepes, A., Loane, R., Rance, B., Lang, F., Ide, N., et al. (2011). A Knowledge-Based Approach to Medical Records Retrieval. *Proceedings of the 20th Text REtrieval Conference (TREC)*.

Duma, D., Liakata, M., Clare, A., Ravenscroft, J., & Klein, E. (2016). Rhetorical Classification of Anchor Text for Citation Recommendation. *D-Lib Magazine*, *22*(9/10).

Garcia Esparza, S., O'Mahony, M. P., & Smyth, B. (2010). On the real-time web as a source of recommendation knowledge. *Proceedings of the fourth ACM conference on Recommender systems* (pp. 305–308). ACM.

Gipp, B., Meuschke, N., & Lipinski, M. (2015). CITREC: An Evaluation Framework for Citation-Based Similarity Measures based on TREC Genomics and PubMed Central. *Proceedings of the iConference 2015*. Newport Beach, California. Retrieved from https://www.ideals.illinois.edu/bitstream/handle/2142/73680/98_ready.pdf

Hofmann, K., Schuth, A., Bellogin, A., & Rijke, M. de. (2014). Effects of Position Bias on Click-Based Recommender Evaluation. *Advances in Information Retrieval* (pp. 624–630). Springer.

Jonnalagedda, N., Gauch, S., Labille, K., & Alfarhood, S. (2016). Incorporating popularity in a personalized news recommender system. *PeerJ Computer Science*, *2*, e63.

Livne, A., Gokuladas, V., Teevan, J., Dumais, S. T., & Adar, E. (2014). CiteSight: supporting contextual citation recommendation using differential search. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 807–816.

Mitzig, N. L., Mitzig, M. S., Martɪnez, F. A., Piriz, R. A., Ferracutti, V. M., González, M. P., & Maguitman, A. G. (2016). SciELO suggester: An intelligent support tool for cataloging library resources. *Library & Information Science Research*, *38*(1), 39–51.

Pan, B., Hembrooke, H., Joachims, T., Lorigo, L., Gay, G., & Granka, L. (2007). In google we trust: Users' decisions on rank, position, and relevance. *Journal of Computer-Mediated Communication*, *12*(3), 801–823.

Phelan, O., McCarthy, K., & Smyth, B. (2009). Using twitter to recommend real-time topical news. *Proceedings of the third ACM conference on Recommender systems* (pp. 385–388). ACM.

Pohl, S. (2007). *Using Access Data for Paper Recommendations on ArXiv. org*. Master Thesis. Technical University of Darmstadt.

Pursel, B., Liang, C., Wang, S., Wu, Z., Williams, K., Brautigam, B., Saul, S., et al. (2016). BBookX: Design of an Automated Web-based Recommender System for the Creation of Open Learning Content. *Proceedings of the 25th International Conference Companion on World Wide Web* (pp. 929–933). International World Wide Web Conferences Steering Committee.

Schwarzer, M., Schubotz, M., Meuschke, N., Breitinger, C., Markl, V., & Gipp, B. (2016). Evaluating Link-based Recommendations for Wikipedia. *Proceedings of the 16th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, JCDL '16 (pp. 191–200). Newark, New Jersey, USA: ACM. doi:10.1145/2910896.2910908

Shelton, B. E., Duffin, J., Wang, Y., & Ball, J. (2010). Linking open course wares and open education resources: creating an effective search and recommendation system. *Procedia Computer Science*, *1*(2), 2865–2870.

Wang, X., Bendersky, M., Metzler, D., & Najork, M. (2016). Learning to rank with selection bias in personal search. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 115–124). ACM.