Searching an Appropriate Journal for your Paper – an Approach Inspired by Expert Search and Data Fusion

Andreas Henrich[1] and Markus Wegmann[2]

[1] Media Informatics Group, University of Bamberg, Bamberg, Germany
andreas.henrich@uni-bamberg.de
[2] markus.wegmann@live.de

Abstract. On an abstract level, one is often confronted with some type of classification problem where we have one example instance or a textual query and we are looking for the class most appropriate for this instance or query. More concretely, we consider journals as classes and the papers published in certain journals as constituting and describing the respective class. In this scenario two information needs are conceivable: (1) We know one paper and we are looking for all journals which could potentially contain similar work. (2) We want to write a paper, have a first working title, and are looking for journals which could be potential targets for a submission of that paper. In this work, we transfer methods used in expert search and data fusion to find appropriate journals: Using a flat, title based search query for articles we examine voting models used in expertise retrieval with its different data fusion techniques to find and rank journals associated with the matching articles that potentially contain most suitable other articles. To evaluate the ranking of found journals, we remove several test articles from the applied collection and utilize them as request items with their titles. We assume that—on average—the journals where these test articles have been published should be among the top ranked journals to provide a suitable result. This fully automated evaluation provides the opportunity to execute a huge number of requests against the collection of articles and to evaluate the different voting techniques transferred from expert search.

Keywords: Expert Search, Expertise Retrieval, IR Systems, Collection Search

1 Motivation and Related Work

One of the approaches introduced for expertise retrieval [1] is based on relevant documents retrieved by a search query. These documents vote for their associated authors as candidate experts. In this work, we transfer this approach to another domain:

Our keyword search on a bibliographic collection yields matching articles which vote for the journals where the articles have been published as beneficial sources or targets. Our aim is to identify a technique, which yields and ranks journals that potentially contain other publications and resources which match the information need of the user. This information need targets journals rather than single articles.

In: M. Leyer (Ed.): Proceedings of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB.
Rostock, Germany, 11.-13. September 2017, published at http://ceur-ws.org

In a first step, we don't evaluate the discussed techniques using manually created test data or test users. Instead, we use article-titles from the collection itself to automatically send these titles as search requests. Since we have the information in which journal j a single article has been published, we can measure the position of this respective journal j in the result ranking and evaluate the algorithms.

This work is based on research in data fusion techniques and their application in the field of expertise retrieval. Different approaches show, that combining multiple retrieval results using voting models can improve retrieval effectiveness [2]. In their survey, Balog et al. [3] present different approaches used in expertise retrieval including the document based voting model. Rank- and score-based fusion techniques are listed and evaluated, mostly based on the work of MacDonald et al. [1]. Furthermore, normalization methods are applied for the underlying candidate expert profiles to gain better results. In the mentioned works, it becomes quite significant that the documents in the upper ranks together with their score values have a disproportionately high impact on the quality of the fusion results; exponential variants of fusion techniques can have better results and prove this fact [3].

In the paper at hand, we investigate how such approaches perform in our setting. We present first promising experimental results and discuss potential future research directions.

It should be mentioned that existing journal recommenders from publishers—like EndNote's manuscript matcher, Elsevier's journal finder, or Springer's journal suggester—are obviously related to our approach. However, these systems apply complex ranking schemes using much more information than our simple approach discussed in this short paper. The aim of our paper is to investigate the capability of rather simple voting techniques in the sketched scenario. A comparison with the existing journal recommenders will be an interesting next step but is out of scope for this short paper.

2 Applied Ranking Techniques

This section describes the utilized ranking techniques. For the flat, title-based article search—i.e. the underlying document ranking—we use Elasticsearch's classic TF/IDF [5] and the BM25 similarity algorithm (cf. section 2.1). For the conversion of the document ranking into a journal ranking—or, more general, collection ranking—four different voting schemes are used (cf. section 2.2).

2.1 Document ranking

TF/IDF. The $score(d,q)$ of a document d given a query q which consists of terms t is computed as follows $score(d,q) = \sum_{t\ in\ q}\left(tf(t\ in\ d) \cdot idf(t)^2 \cdot norm(d)\right)$. The term frequency tf of term t in document d is computed as $tf(t\ in\ d) = \sqrt{frequency}$. The inverse document frequency idf for term t is computed as $idf(t) = 1 + \log\left(\frac{numDocs}{docFreq(t)+1}\right)$, where $numDocs$ is the number of all documents in the collection and $docFreq(t)$ is the number of documents containing term t.

The normalization factor $norm(d) = \frac{1}{\sqrt{numTerms}}$ for a matching document d causes higher weights for short documents (documents with a lower number of terms $numTerms$) in the score computation.

The score formula for Lucene's classic similarity TF/IDF in addition contains other weighting factors (for normalization and coordination) which are not considered or not relevant in our experiments and not involved in the score-computation.

BM25. For BM25, the $score(d, q)$ of a document d given a query q which consists of terms t is computed as follows:

$$score(d, q) = \sum_{t\ in\ q} \left(idf(t) \cdot \frac{tf(t\ in\ d) \cdot (k+1)}{tf(t\ in\ d) + k\left(1 - b + b \cdot \frac{|D|}{avgdl}\right)} \right)$$

The term frequency tf describes the number of occurrences of term t in document d, $|D|$ represents the document length, and $avgdl$ is computed as the average document length over all documents in the collection.

Here the inverse document frequency idf for term t is computed as $idf(t) = \log\left(1 + \frac{numDocs - docFreq(t) + 0.5}{docFreq(t) + 0.5}\right)$, with $numDocs$ and $docFreq(t)$ defined as before. In our experiments, we use BM25 with standard values for k (1.2) and b (0.75).

2.2 Collection ranking

Based on the article ranking as search result, four approaches introduced in expert search are adopted to derive a journal ranking—respectively, a collection ranking. In general, the voting model can be based on different inputs: the number of items in the search result associated with a collection, the ranks of the items associated with a collection, and the score values calculated for the items associated with a collection [1].

Let $R(q)$ be the set of articles retrieved for the query q and $score(j, q)$ the computed score for journal j and query q, we apply four different voting models:

Votes. This metric takes the number of found articles for every journal as the score:

$$score_{Votes}(j, q) = |\{article \in R(q) \cap Journal(j)\}|$$

CombSUM. For every journal, this aggregation sums up the scores of the articles:

$$score_{CombSUM}(j, q) = \sum_{article\ \in\ R(q) \cap Journal(j)} score(article, q)$$

CombANZ. This aggregation is based on the previous CombSUM method normalized by the number of found articles for every journal (i.e. divided by the Votes-value):

$$score_{CombANZ}(j, q) = \frac{score_{CombSUM}(j, q)}{score_{Votes}(j, q)}$$

CombMAX. This metric takes the first result stemming from j, respectively, the article with the highest ranking, as voting candidate with its score:

$$score_{CombMAX}(j,q) = Max(\{score(article,q) : article \in R(q) \cap Journal(j)\})$$

3 Experiment

For our experiments, we take data from the dblp computer science bibliography (Digital Bibliography & Library Project), an online reference for bibliographic information on major computer science publications [4]. Dblp offers bibliographic metadata and links to the electronic editions of publications and consists of nearly 3,800,000 publications. The data is published by the University of Trier which exposes a dump for download and further research. In this work, we restrict the experiments using only the roughly 1,500,000 articles published in the 1,657 journals ignoring e.g. conference articles.

In the first step, we take 10,000 randomly chosen articles and remove them from the collection. All removed items serve as requests for the evaluation series with their title. The length distribution of all titles used as keyword queries is as follows: The minimum is 1 word. 25% of the queries/titles comprise 7 or fewer terms. The median is 9 and the 3rd quartile is 12. The longest title among the 10,000 chosen titles consist of 37 words.

For every item of the 10,000 articles, all combinations of the two similarity algorithms and the four voting models described in section 2 are combined and applied. Hence, every search request yields eight sets of ranked journals.

For each article and for each set of ranked journals the rank of the article's corresponding journal is determined and saved. Ideally, an article's corresponding journal should appear at rank 1 or at least among the top ranks of all ranked journals.

Our setup contains 1,657 different journals and nearly 1,500,000 articles. Figure 1 shows the distribution of articles over journals. Nearly 60% of all journals (965 items) contain 500 articles at most. As the maximum, the collection contains one journal in which more than 18,000 articles are published.

4 Results

For all 10,000 article-requests, we determine the rank of the article's corresponding journal. Table 1 shows the effectiveness results for each combination of *document-similarity model* and *voting technique*. We state the 1st quartile (value x meaning that for 25% of the queries the respective journal has been among the top x results), the median, the 3rd quartile, the share of queries where the expected journal is in top 10, and the mean reciprocal rank (MRR).

Aggregations calculated by the Votes algorithm for BM25 and TF/IDF have the worst—and the same—results. For only 25% of the queries the journal associated with the requested article appears among the top 30 results. CombANZ and CombSUM follow, they deliver at least 25% of the corresponding journals within the top 20 results.

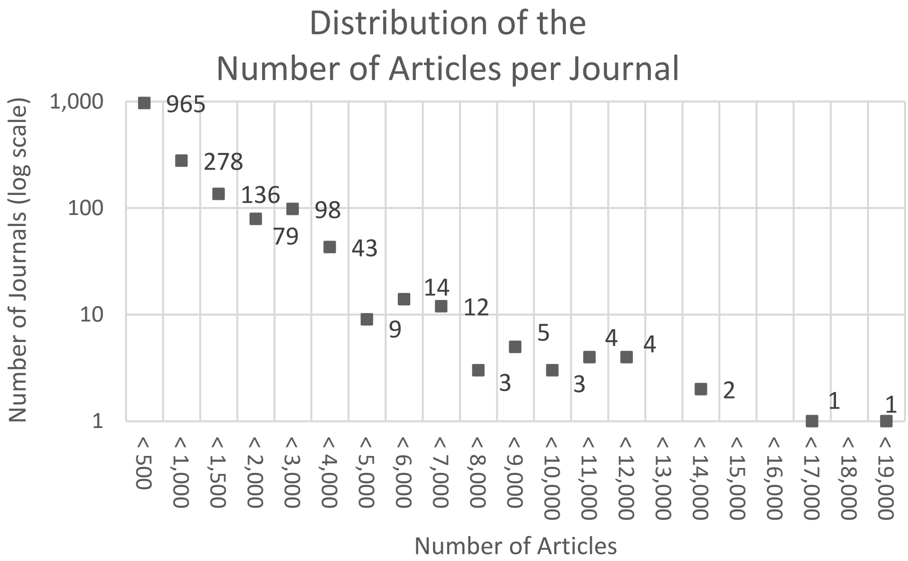

Fig. 1. All 1,657 journals categorized by the number of published articles

model ►	BM25				TF/IDF			
voting technique (combination) ►	CombANZ	CombMAX	CombSUM	Votes	CombANZ	CombMAX	CombSUM	Votes
1st quartile	19	3	16	30	16	3	12	30
median	70	12	72	117	57	14	58	117
3rd quartile	242	59	239	336	193	67	202	336
share with journal in top 10	15.7%	48.5%	20.5%	13.5%	18.4%	44.9%	24.0%	13.5%
MRR	0.06	0.27	0.1	0.07	0.07	0.25	0.13	0.07

Table 1. Effectiveness measures for the rankings of the articles' corresponding journals

The best results for getting the associated journal are achieved by applying CombMAX which returns the searched journal among the top 12, respectively top 14, journals in 50% of all cases.

The presented results suggest that the underlying similarity algorithms (retrieval model) do not fundamentally change the ranking behavior of the imposed collection ranking methods. A much stronger influence can be observed for the combination schemes. Hence, these schemes seem to be worth more in-depth consideration.

5 Conclusion and Future Work

When comparing all ranking methods, it turns out that CombMAX yields the best results regarding the journal ranking. Here, no noise based on lower ranked results—respectively, articles—that contribute as voters is generated. In certain cases, these lower ranked results considered in the aggregation distort the ranking. When applying CombANZ and CombSUM, this effect is shown. Both methods yield worse results, especially regarding the rankings above the median and first quartile.

The approach using the Votes algorithm is not well suited in this form: not even 50% of the rankings see the expected journal under the top 100. This is because the approach does not differentiate between results on higher and on lower ranks.

CombMAX as the best performing approach is also in accordance with the experimental results presented in [1], when no profile length normalization was applied. Nevertheless, in our perception the results achieved by the simple CombMAX approach are surprising since the approach corresponds to a nearest neighbor based classification or a single link approach.

For further research, we plan to modify the applied aggregation techniques: according to current results, a CombSUM technique which considers only the upper ranking articles might deliver more accurate results. Considering only the top article result for every journal would end up in the applied CombMAX method.

Taking a closer look at the results it is a bit astonishing that CombSUM is much closer to Votes than to CombMAX. One could hypothesize that the decline of the score values might be well suited for ranking single documents but not for summing them up to yield a combined score for a journal. Problems with independence assumptions as well as ranking equivalent transformations in the formulas might be potential reasons in this respect. We plan to consider this further and to investigate combination schemes reflecting these observations. The sketched observations might also be the reason why using the reciprocal ranks from documents—respectively, articles—in combination schemes performed surprisingly well in [6].

6 References

1. Macdonald, C., Ounis, I.: Searching for Expertise: Experiments with the Voting Model. In: THE COMPUTER JOURNAL, Vol. 52 No. 7, pp. 729–748. Published by Oxford University Press on behalf of The British Computer Society 2008.
2. Lee, Joon Ho: Analyses of Multiple Evidence Combination. In: SIGIR 97 Philadelphia PA, USA, pp. 267-276.
3. Balog, K., Fang, Y., de Rijke, M., Serdyukov, P., Si, L.: Expertise Retrieval. In: Foundations and Trends in Information Retrieval, Vol. 6, Nos. 2–3, pp. 180–184, 2012
4. dblp Homepage, http://dblp.uni-trier.de/, last accessed 2017/06/12.
5. Elasticsearch homepage, https://www.elastic.co/guide/en/elasticsearch/guide/current/practical-scoring-function.html, last accessed 2017/06/12.
6. Macdonald, C., Ounis, I.: Voting for candidates: Adapting data fusion techniques for an expert search task, in Proc. of the ACM International Conf. on Information and Knowledge Management, (CIKM '06), New York, NY, USA, pp. 387–396, 2006.