Expert Search

Yulia Gurova (✉), Ilya Makarov, and Leonid E. Zhukov

National Research University Higher School of Economics, Moscow, Russia
ygurova@hse.ru, iamakarov@hse.ru, lzhukov@hse.ru

Abstract. We present the expert search system for a list of keywords as a query. The recommendation is based on the graph of coauthors and citations built on publications of the search field.

Keywords: Social Networks; Machine Learning; Expert Search

1 Introduction

Expert search (ES) is a task of detecting a person with highest expertise in some scientific field. It arises in a wide range of applications. First, ES may be used for the automation of distribution of proposed publications between the revisers by the journals and could make it more effective. For this purpose it should be an expert from the same narrow field as the author, even if he is not very famous and cited. At the same time, for the second application — the search of expert as an adviser for some particular problem — it should be an expert in wide field, so that he could name the best method even though he doesn't know in details how it works. We must separate these tasks as they usually need different approaches.

There are two main approaches that are commonly used in computational-based expert search (their combinations are used as well). The first one is Profile-Centric Method. It is based on information about the researcher, his history of publications, positions in institutes and universities, academic supervision, conferences talks and so on. The second one, Document-Centric method is based on the bibliographic data, and detects expert using his publications' features mostly: the key words, journals, citations and so on. Then the authority of the researcher as an expert is measured as the quality of publications through the journal rankings, number of citations and so on. The second one is considered as the more effective one for solving the expert search in academia due to its speciality [6]. Also, as the systems of paper recommendations are now highly developed, it may be used in ES.

There are some models of the research space, that store and analyse articles from different fields. Such as Google.Scholar, Scopus, Web of Science, Microsoft Academics and so on. All of them provide search services for articles, but not for the experts.

Due to the data restrictions (which are described in section 3 in our project, the second variant was chosen, and our system uses data on publications for expert recommendation. There are several approaches for ES built on the bibliographic data. The most popular approaches are based on the full text of article as a query for expert search.

The paper is organised as follows. The existing approaches are described in section 2, our model is presented in section 4, the used dataset is described in section 3, section 5 presents the results, section 6 concludes.

2 Related Work

In this section some systems that use the publications data for Expert Recommendation are discussed.

The recommendation system build on the big scholarly data is described in [5]. The system outputs three types of recommendations: Expert, Classic and Serendipity for the paper as a query. All three are based on bibliographic data. For the Expert recommendation the closest paper at the very specific, narrow topic of the queried paper is found. The experts are the authors of this paper. The Classic includes papers from the same field, but wider range, and focuses more on the rankings of the papers (gives the ones, that are changing and most cited in the field). This may help to dive into the field to the newcomer. Serendipity weights highly the newest papers and last breakthroughs and should be used by scientists from the same field to follow the latest news.

The mechanism for Expert recommendation for this system is following: the article-level Eigenfactor (ALEF) is calculated for the paper from the query. Then the closest paper is found in the data. The author of the closest paper proposed to be the best expert for the given paper.

The ES may be considered as a part of a larger field — topic modeling. The most used modern approaches here are text embeddings and probabilistic methods based on the probabilities of appearing of some concrete word in some topic. The rare words then have more weight, and summarising these probabilities for all the words in a text (weighted on frequency), we can find the probability for this text to be from any topic. Having a range of such topics, we find the one, for which the probability is the highest, to determine the topic. These methods are developed in different other tasks of NLP.

Topic modelling is based on NLP. There are methods (based on NLP) of embedding texts to some space of topics, where close in topics text has shorter distance. Building such space is a complicated problem as the amount of information is enormous and the number of articles is growing up speedily. In the lack of resources the abstracts of articles may be used in the same way, but the results will be less precise. In our model we used the key words formulated by authors as the description of the topic without embeddings.

Based on these models the expert recommendation may be implemented in the following way. Having a paper, for which the reviewer is necessary, we determine the topic of the text, and then take the expert from the topic. As the topics should be determined in advance, the expert may be chosen in advance for every topic from the authors articles on this topic.

The models first used in these field are PLSA (probabilistic latent semantic analysis) [2], based on LDA (latent Dirichlet allocation) [1] models, SWB (special words with background) based on noised PLSA, and EM-algorithms which were modified by Vorontsov to build system called BigARTM (Additional Regularisation TM [4].

Another approach of topic modeling is emending. There is no need in determination of concrete topics.Texts which are close in topic are projected closely in distance to some space. Then the positions of the experts may be measured in the same space based on his publications. To determine the expert for an article, this article is projected in the topic-space, and then the closest expert is found.

ArnetMiner [3] is a system that was developed by Chinese researches in 2008. It aims to model the science fields and search most cited and important articles in the field, expert search and other recomendations. It accumulates all kind of information about authors, publications and conferences. Also, they search for the connections between the researches: co-authorship, citations, academic supervision, co-working, subbordination positions, family connections.

[**?**] contains more detailed description of expert search of the original ArnetMiner system.

The current code for AMiner is available on GitHub since the June 2019. The module for expert search uses FastText to find the closest expert to the downloaded text. First, the FastText model is trained on the database with all articles and authors' fields. Then the closest author for the text is found by the L2-distance. The requested number of the closest authors is returned by the module.

The same model could be used for our data, but we have only abstracts of the articles, which may not be enough for correct results. Moreover, in the dataset there is much larger amount of authors and articles.

3 Dataset

The data was provided by the scrapers team of the project. Two datasets were used. First contains data on authors, there articles, coauthors and some personal features like institute/organisation, city and so on. These features are provided for less then 5 persent of authors, so they can not be useful for the model. The second dataset contains articles features: authors, keywords, journal and abstract.

For every author the *'field'* feature was filled by key words of his articles. New feature was added: the dictionary with the counts of key words. These features are necessary for the ES model.

The data significantly differs from the one used by AMiner, which makes impossible the use of their model and requires the development of our own. The comparison is shown in table 1. Using the network properties we may focus only on the connections between the authors in particular field of interest.

There are some personal features in out data, but it is known for a negligibly small amount of authors, so they may not be used in the model. The feature

System	# of articles	# of researchers	# of conferences	personal features
AMiner	185 725 922	113 152 022		yes
Our project	1 610 955	1 300 713	8525	no

Table 1. Datasets comparison to AMiner

'field' may be refilld by our function if necessary. In this case it fills it with the list of tags of the authors publications. Also the new features appears, the dictionary with the counts of number of appearances of all these keywords in authors articles. If we choose an expert, we should be more interested in an author, who has more publications in the field, keeping all the rest fixed.

Field	# of missings	%of missings
location	1 246 757	99.8
organisation	1 234 744	99.8
citationNumber	1 245 438	99.7
h-index	1 245 356	99.7
field	927 259	74.2

Table 2. Missings counts

4 Model Description

Due to data and computational restrictions, the model was built using the information about publications. It takes the .json files as input. The files contain information about authors, articles and conferences. As the data on articles and authors is not really connected to the data on conferences, the last one is imported but not used in the current version. It is planned to be important in the future versions.

The algorithm is the following. The main function uses the data to build a citation graph of the authors, who published the articles with the keywords containing the tags, and the authors, who they cited. Then, using the PageRank algorithm it detects the most influential people in this graph and returners the ranked list of names of the experts, together with their ids in the system and pagerank in the graph.

Algorithm 1: How to write algorithms

```
Input: List of articles  with keywords
Result: experts = the ranked list of experts
initialization;
```

Also, the system has a branch of helpful intermediate functions. It returns the necessary info about the author by authorID in the system or by name. It returns info about the articles by id in the system or by DOI. In fact, there are some articles in the authors data, which are not presented in article data, so we can not get the details. But for some of them we can get the DOI and coauthors from the authors dataset.

One more intermediate function is the citations graph which may be used for some other purposes. Also the co-authorship graph for the authors from some field is built by the system. For the given data it turned up to be futile. The results, taken from that graph using centrality measures are not comprehended as the authors with a lot of connections were not the real experts in the field. The citations graph gives much better result for this data. But if necessary, the co-authorship network may be used for expert search using other database.

5 Discussion

The model was checked using existing system (AMiner) and common sense as a benchmark. Here we describe one example to see how it works. The chosen field is 'Computer vision' and it was the input tag. The citations digraph built for this field had 7182 nodes and 36086 edges. 10 experts with highest PageRank in this network are Reiner Lenz, David G. Lowe, Andrew Zisserman, Zhengyou Zhang, John F. Canny, Alex Pentland, Cordelia Schmid, Tomaso A. Poggio, Anil K. Jain, Takeo Kanade, Richard Szeliski.

The first expert in AMiner for this field is Tomaso A. Poggio, who is 8th in our list. The second in AMiner's ranking is Andrew Zisserman, who is the 3d in our list. Takeo Kanade is in first 10 in both rankings. Alex Pentland goes in the second decade by AMiner, and Zhengyou Zhang appears as a coauthor of one of first experts in AMiner's ranking. All the uther people have cited publications in the field, so the model gives sensible results.

There are a lot of ways to develop this model. The most obvious and, probably, most effective is to use some personal features in the search like Hirsch index.

6 Conclusion

In this paper the existing approaches to Expert Search are discussed. Then the new developed system is described and the results are analysed. The existing approaches are mostly based on probabilistic methods and text embedding and usually require the full text of an article to find the closest expert. The described system uses data on publications and citations network to detect the experts and gives sensible results.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research 3(Jan), 993–1022 (2003)

2. Hofmann, T.: Probabilistic latent semantic analysis. In: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. pp. 289–296. Morgan Kaufmann Publishers Inc. (1999)
3. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks pp. 990–998 (2008)
4. Vorontsov, K., Frei, O., Apishev, M., Romov, P., Dudarenko, M.: Bigartm: Open source library for regularized multimodal topic modeling of large collections. In: International Conference on Analysis of Images, Social Networks and Texts. pp. 370–381. Springer (2015)
5. West, J.D., Wesley-Smith, I., Bergstrom, C.T.: A recommendation system based on hierarchical clustering of an article-level citation network. vol. 2, pp. 113–123. IEEE (2016)
6. Xia, F., Wang, W., Bekele, T.M., Liu, H.: Big scholarly data: A survey. vol. 3, pp. 18–35. IEEE (2017)