CSSeer: an Expert Recommendation System based on CiteSeerX

Hung-Hsuan Chen[†], Pucktada Treeratpituk[‡], Prasenjit Mitra[†‡],C. Lee Giles[†‡]
[†]Computer Science and Engineering, [‡]Information Sciences and Technology
The Pennsylvania State University, University Park, PA 16802, USA
hhchen@psu.edu, pxt162@ist.psu.edu, pmitra@ist.psu.edu, giles@ist.psu.edu

ABSTRACT

We propose CSSeer[1], a free and publicly available keyphrase based recommendation system for expert discovery based on the CiteSeerX digital library and Wikipedia as an auxiliary resource. CSSeer generates keyphrases from the title and the abstract of each document in CiteSeerX. These keyphrases are then utilized to infer the authors' expertise. We compared CSSeer with the other two state-of-the-art expert recommenders and found that the three systems have moderately divergent recommendations on 20 benchmark queries. Thus, we recommend users to browse through several different recommenders to obtain a more complete expert list.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.7 [**Information Storage and Retrieval**]: Digital Library—*Collections, Dissemination*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*relevance feedbacks, retrieval models, selection process*

Keywords

Expert Recommendation, CSSeer, CiteSeerX, Information Extraction, Text Mining, Keyphrase Extraction

1. INTRODUCTION

We introduce the data analysis flow of CSSeer framework. The keyphrases of each document and an expert's list of phrases are analyzed and indexed offline. When a user inputs a term for which an expert is sought, the query processor communicates with the indexed expert list and related keyphrases online.

From CiteSeerX corpus, the field analyzer extracts the title, abstract, published venue name, reference list, author names, author emails, and author affiliations for each document using ParsCit [1]. A Support Vector Machine based header parser is utilized to extract other metadata. The author names are disambiguated by Random Forests [2].

CSSeer harvests the titles and the hyperlink texts from the introduction paragraphs of Wikipedia pages related to computer science, statistics, and mathematics. These phrases are used to compile keyphrase candidates. To increase recall, CSSeer extracts bigrams, trigrams, and quadgrams appearing at least 3 times in titles of papers in CiteSeerX as keyphrase candidates. Since the titles and hyperlink texts in Wikipedia are edited by users, they are typically high quality phrases with meaningful semantics. To identify keyphrases for each CiteSeerX document, CSSeer compares the title and abstract of each document with all the Wikipedia keyphrase candidates and uses the matched terms as keyphrases.

CSSeer discovers experts of a given term by considering both textual relevance and quality (inferred by citation counts) of the authors' published papers. The expert score of an author a to a query term q is defined as follows.

$$p(a,q) := \sum_{\forall d \in pub(a)} \log(c(d)) \left(tf(q) / \sum_{\forall q' \in d} tf(q') \right), \quad (1)$$

where $pub(a)$ returns a set of publications of author a, $c(d)$ returns the citation counts of the document d, and $tf(q)$ returns the term frequency of q in d.

2. COMPARISON OF DIFFERENT EXPERT RECOMMENDERS

Manually evaluating an expert recommendation system requires the evaluators to have sufficient domain knowledge to judge the quality of the recommended expert list. Recruiting knowledgable evaluators is not easy, and asking the evaluators to label the recommended list is time consuming. Instead, we compared the expert list returned by CSSeer with the other two state-of-the art expert recommenders. We define consensus score $S@n$ of one expert recommendation system e_i to the other systems $e_1, \ldots, e_{i-1}, e_{i+1}, \ldots e_n$ in Equation 2.

$$S@n := \left| \bigcup_{\forall k \neq i} \left(r_i^{(n)} \cap r_k^{(n)} \right) \right|, \quad (2)$$

where $r_i^{(n)}$ is the set of the top n returns of the recommendation system e_i, and the $|\cdot|$ function returns the set length.

The consensus score involves no user evaluation and as such could be automated for a large number of queries. However, different expert recommendation systems record

[1] http://csseer.ist.psu.edu/

JCDL'13, July 22–26, 2013, Indianapolis, Indiana, USA.
ACM 978-1-4503-2077-1/13/07.

Table 1: The consensus score@10 and consensus score@20 of 20 benchmark queries among the returns of CSSeer, ArnetMiner, and Microsoft Academic Search (MAS).

	$n = 10$			$n = 20$		
Query	CSSeer	ArnetMiner	MAS	CSSeer	ArnetMiner	MAS
algorithm	1	2	1	5	4	3
security	3	4	1	12	10	9
software engineering	3	3	4	11	9	11
information retrieval	6	7	5	14	15	12
machine learning	2	3	1	8	8	9
database	4	6	6	9	10	14
programming language	4	3	5	9	7	10
data structure	6	4	5	12	8	13
world wide web	4	5	6	9	11	14
social network	2	2	3	5	5	6
compiler	4	4	3	12	12	13
vlsi	1	1	1	2	2	1
computer network	1	1	0	2	4	2
support vector machine	1	4	4	9	12	8
semantic web	7	7	7	11	13	14
nonparametric statistics	2	2	0	2	4	2
markov chain monte carlo	4	0	4	9	2	9
quality of service	5	2	5	8	6	6
virtual machine	4	3	1	6	6	4
intelligent agent	5	8	6	9	12	10
Average Consensus Score	3.45	3.5	3.5	8.2	8.0	8.5
Standard Deviation of Consensus Score	1.82	2.09	2.28	3.53	3.83	4.35

the same expert with different name variations. For example, W. Bruce Croft at the University of Massachusetts is recorded as "W. Bruce Croft" in both CSSeer and MAS but is "Bruce Croft" in ArnetMiner; ChengXiang Zhai at UIUC is stored as "ChengXiang Zhai" in both CSSeer and ArnetMiner but is "Cheng-xiang Zhai" in MAS. Therefore, naïvely regarding names as strings and performing string matching could generate misleading results.

As a result, we intentionally selected 20 benchmark diverse queries that contain both broad and narrow topics, including hardware ("vlsi"), low level machine concepts ("compiler" and "virtual machine"), software development ("programming language", "data structure", and "software engineering"), statistical techniques ("nonparametric statistics" and "markov chain monte carlo"), data mining techniques ("information retrieval", "support vector machine", etc.), and so on. We did not use the relevant judgements provided by ArnetMiner directly[2] [3], because there these terms are mostly limited to the artificial intelligence domain.

We compared S@10 and S@20 for the three systems for 20 benchmark queries. The results were manually examined and reported in Table 1. Note that in the table the name variations for different systems were unified for better representation. As can be seen, the average consensus scores S@10 and S@20 are low for all three systems. Specifically, on average only 3.45 to 3.5 names out of the top 10 returned by one system overlapped with at least one of the other two systems. For the top 20 returns, the numbers of overlapping names are still small, on average ranging from 8.0 to 8.5. This suggests that the current state-of-the-art expert recommendation systems have divergent opinions. Relying on only one expert recommendation system may produce a biased expert list.

Although we try to select benchmark queries that cover several different domains in Computer Science, the number of queries is still small. A large scale experiment is limited by the evaluation process which requires users with domain knowledge. Although it is possible to automatically compare the overlap of top-n returns across different systems, the name variations can still be an issue in practice. For future work, we are planning to use regular expression and edit distance to compare the same names in different systems. Such a system could automatically compare the consensus of different expert recommendation systems. Currently, the expert scores are calculated offline on a fixed keyphrase list. However, users might submit a valid term which is not included in the pre-defined keyphrase list. For this one could integrate the search tool Solr/Lucene for fast document lookup and online expert score approximation.

[2] `http://arnetminer.org/lab-datasets/expertfinding/`

Acknowledgments

We gratefully acknowledge partial support by the National Science Foundation and Dow Chemical.

3. REFERENCES

[1] I. Councill, C. Giles, and M. Kan. Parscit: An open-source crf reference string parsing package. In *Proceedings of LREC*, volume 2008, pages 661–667. European Language Resources Association (ELRA), 2008.

[2] P. Treeratpituk and C. Giles. Disambiguating authors in academic publications using random forests. In *Proceedings of the 9th ACM/IEEE Joint Conference on Digital Libraries*, pages 39–48. ACM, 2009.

[3] Z. Yang, J. Tang, B. Wang, J. Guo, J. Li, and S. Chen. Expert2bole: From expert finding to bole search. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,(KDD'09)*, pages 1–4, 2009.