A BFS-BASED SIMILAR CONFERENCE RETRIEVAL FRAMEWORK

Qing Guo[1, 2]

[1]Nanyang Technological University, Singapore
[2]SAP Innovation Center Network,Singapore

ABSTRACT

Literature review is part of scientific research. Online references management tools help researchers in finding relevant literature and documents. Finding relevant conferences is the key step to understand the research field. Researchers usually rely on the conference names to find out whether they are related. However, the conference name rarely reflects the diverse topics it covers. For instance, for the two conferences, "International Conference on Data Mining and Applications" and "Special Interest Group on Information Retrieval" which represent similar research topics and research areas, but the names fail to capture the similarity. One possible method to compute the similarity between all the papers in the two conferences but it's time-consuming. Instead of computing the similarity, this work builds a search engine based on Lucene and find similar conferences given a query conference based on the index. A BFS-based algorithm is proposed to address this problem and experiments on DBLP dataset shows the proposed approach can generate comparable results with the similarity-based approach.

KEYWORDS

Literature Review, Information Retrieval, Breadth-First-Search

1. INTRODUCTION

The ability of retrieving relevant information is of fundamental importance in the big data era [2]. Google, Yahoo, Bing or Baidu are popular search engines. With massive amount of data owing into the search engine, efficiency and scalability are key factors that influence the performance of the system and user experience. In this paper, searching similar conferences or journals are studied. Among the activities in the literature review, finding relevant conferences and journals is a critical step which helps the author gain an overview of the research field. A greedy approach is to get a ranking list of top-N conferences based on the similarity of the query conference and the others. In this way, the papers of the conferences need to be aggregated into one document to facilitate the similarity computation. Nevertheless, the large amount of papers makes this approach inefficient in both processing speed and storage. This work aims to address these two issues by leveraging index structure built by Lucene. Firstly, a IR system is developed for computer science publication based on the DBLP[1]dataset. An efficient algorithm based on BFS (Bread-First-Search) is developed to discover the top-N most similar conferences to query (publication venue and year, e.g., "SIGIR+1990").

[1]http://dblp.uni-trier.de/faq

David C. Wyld et al. (Eds) : CSITY, DTMN, NWCOM, SIGPRO - 2018
pp. 53–60, 2018. © CS & IT-CSCP 2018 DOI : 10.5121/csit.2018.81105

2. FRAMEWORK

2.1. Dataset

DBLP is a computer science bibliography website that contains more than 2.6 million of documents, published by more than 1.4 million authors stored in the form of XML. Considering the multiple dimensions for each record, it is impossible to use a DOM XML parser to infer the required attributes since the parse tree would become too big to fit the memory. SAX[2] (Simple API for XML) is utilized for parsing XML file. The information fields are extracted by the parser including authors, title, venue, year, type and paper id. In this work, the articles and the conference that compose about the 93.8% of the entire dataset are further used for implementation of the algorithm.

2.2. Lucene

Moreover, recent version of Lucene 5.0.0 supports some pre-processing/cleaning step like tokenization, stemming or lower-case normalization Lucene[3] is a project maintained by the Apache foundation which offers a complete set of APIs for building a search engine and is applied in many commercial systems. Lucene provides a complete set of tools to perform document indexing and search/ranking operations.

2.3. Indexing

To index the paper information, several operationsare conducted by Lucene:

- Extract paper information from the XML file.
- Lucene Analyzer normalizes the extracted text in different fields through tokenization, stemming, lower case and stopword removal. The tokens are saved in the document.
- The inverted index is constructed by the IndexWriter.

By indexing, users can input keywords in the various fields to trigger the search. For instance, the user can input a conference as a query (venue: SIGIR, Year: 1991) and the system will trigger the search based on this query.

3. SEARCH SIMILAR CONFERENCES

This task can be formulated as: given the paper titles of a certain conference as a query, e.g., SIGIR 2010, search 10 conferences that are most similar to the query conference. Document similarity is an extensively studied topic in text mining area. Various similarity calculation methods have been proposed including Jaccard Index, Levenshtein,N-Gram distance, etc. In this task, we propose to exploit N-Gram distance to capture the similarity between two conferences.

3.1. Similarity Computation

N-Gram model is widely applied in text mining. In the fields of computational linguistics and natural language processing, an n-gram is a contiguous sequence of n items froma given sequence of text or speech. N-Gram model has been used in a variety of NLP tasks, such as spelling

[2]http://www.saxproject.org/
[3]http://lucene.apache.org/

correction, word breaking and text summarization. Another application of N-Gram model is for extracting features for supervised Machine Learning models such as SVMs, Naive Bayes, etc. N-grams are basically a set of co-occurring words within a given window and when computing the n-grams, we typically move one word forward. For example, given a sentence "I like the information retrieval course", when $n = 2$(also known as bigrams), then the n-grams can be generated as: "I like", "like the", "the information", "information retrieval", "retrieval course". In this task, we use N-Gram distance [1] to compute the similarity between two conferences ($sim(c_1, c_2)$). Nevertheless, the details about this technique would not covered in this report. To implement N-Gram distance, we use N-Gram Distance class in Lucene library.

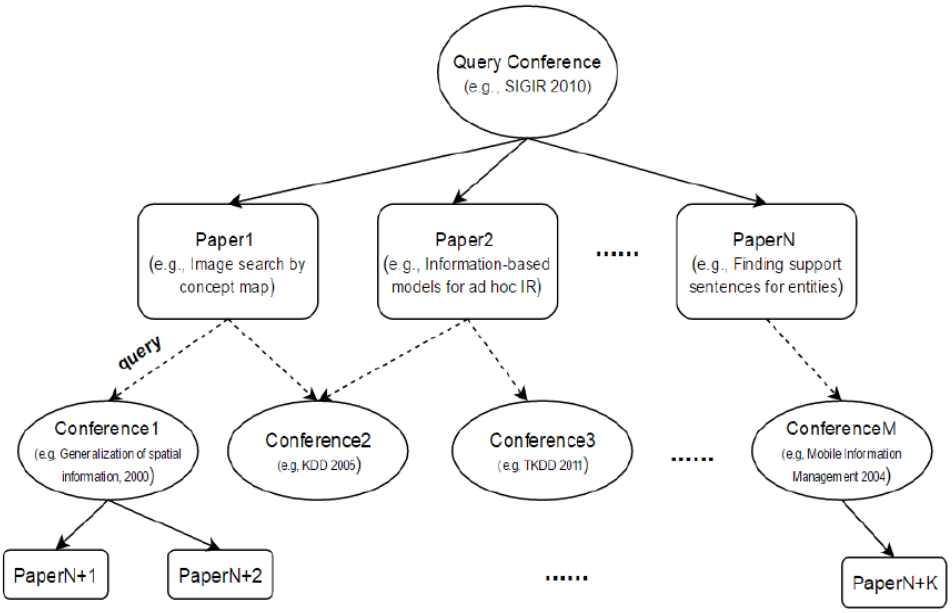

Figure 1. Tree structure of conferences and papers

Table 1. Time required for the tasks.

Task	Average time
Retrieve all paper titles of a conference	$2350ms$
Compute similarity of two conferences	$50ms$
Obtain top 10 similar conferences of a given conference	$(250 + 50) \times \frac{21000}{1000} = 6300s$

3.2. A BFS-based Similar Conferences Search Algorithm (BSCS)

An intuitive idea to search top N similar conferences is greedy search. We attempt to analyse the feasibility of greedy search by simply calculating the following statistics in Table 1. Random queries are selected to estimate the total time. For the first task, we randomly pick 20 conferences to retrieve all the paper titles of each one using the built system and each retrieval averagely takes about $250ms$. Next, 100 random conference pairs are generated, e.g., <SIGIR 1990, Pictorial Information Systems 1988>, then we apply *N Gram Distance* to them and finally the average time is $50ms$. In total, there are nearly 21000 conferences, thus for a single conference, it may take more than$1.5h$ (not include ranking) to find top 10 similar conferences, making it impossible as

an efficient application. In this section, we propose a highly efficient method to address this task, named as a BFS-based similar conference search algorithm shown in Algorithm 1 and Figure 1. The searching component is triggered by the query conference of which all the paper titles, $c.papers$, are returned. Then the paper titles are taken as keywords to find similar papers, P, based on the index built by Lucene. We assume if two papers are similar, their conferences are also relevant. Hence, the conferences of these similar papers are further retrieved as $p'.conf$. The similarity of retrieved conferences and query conferences are measured by N-Gram distance which is maintained by a priority queue, Q_1. A variable $scanNum$ is set to update the number of similar conferences. If this number exceed N, then the searching stops and return the top-N elements of Q_1.

Algorithm 1 *A BFS-based Similar Conference Search Algorithm (BSCS)*

Input: A query conference c_q, and N, the number of similar conferences
Output: Top-N most similar conferences

```
Initialize an empty priority queue Q1 and an empty queue Q2, scanNum = 0 and c=c_q
while scanNum < N do
    for p in c.papers do
        Obtain papers P by using p as a query
        for p' in P do
            if Q1.has(p'.conf) then continue
            Q1.push(p'.conf, sim(c_q, p'.conf))
            Q2.push(p'.conf)
            scanNum+ = 1
    while c has been scanned do
        c ← Q2.pop()
return top-N conferences in Q1 as a list
```

Figure 2. A BFS-based Similar Conference Search Algorithm (BSCS)

3.3. Performance Analysis

3.3.1. Efficiency Analysis

Note that the main motivation of our proposed algorithm is retrieving top 10 similar conferences at an acceptable speed. Though there are other approaches for this purpose we also considered, such as clustering documents by either TF-IDF features or topics. In this direction, given a query conference and its cluster label, we could only take the conferences belonging to the same cluster as candidates. However, this approach would be extremely time-consuming and require high-performing machines to provide high computation power (it has taken us more than 2 days for an unfinished clustering task), forcing us to give up this method.

Without compromising the quality of results, we take an economic strategy that make the full advantage of already built index by Lucene. Having tested 100 queries, the average time is about 45 seconds (ranging from 8*s* to 75*s*) which could be acceptable for users due to the large-scale data to process. Meanwhile, an additional advantage of our algorithm is that it does not rely any intermediate results from other algorithms (e.g., topic model or clustering), saving the system huge amount of storage and reducing the complexity of this job.

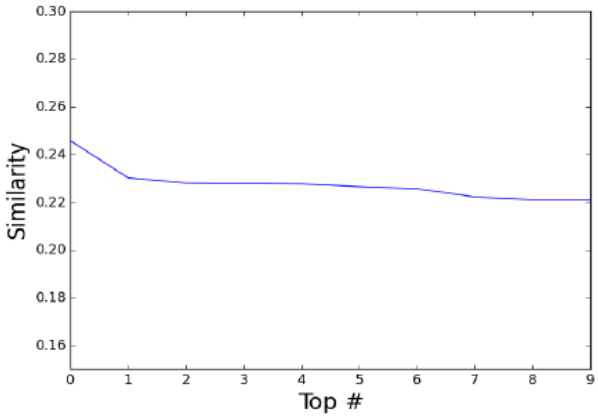

Figure 3. Similarity between Top 10 results and "SIGIR 1990"

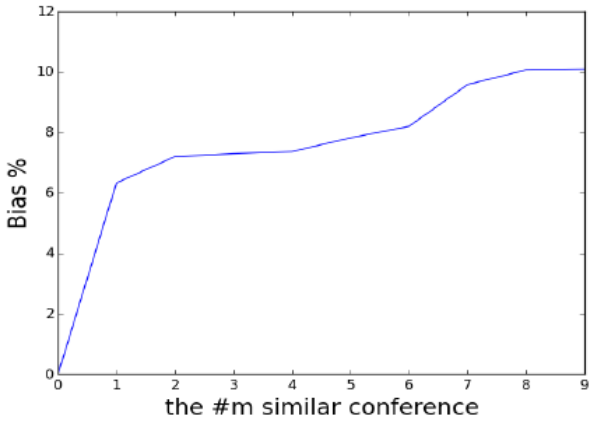

Figure 4. "SIGIR 1990": Difference between R_m and "SIGIR 1991"

3.3.2. Effectiveness Analysis

The effectiveness of BSCS will be evaluated from two perspectives: quantitative and qualitative evaluation.

- **Quantitative evaluation.**

Several query conferences from various research areas are tested and almost all the similarity lies in the range from 0.20 to 0.25. "SIGIR 1990" is chosen as an example presented in Figure 2. However, the evaluation only relying on similarity value is unfair to demonstrate the effectiveness of BSCS since similarity itself depends on the data quality. Thus, we conduct comparison analysis that can ease this issue. Since it would take much time (>1.5h) to retrieve the top 10 similar conferences given a conference by greedy search, we adopt an assumption that two conferences ($C_{v,i}$ and $C_{v,i+1}$) published in the same venue (v) and in consecutive years (i and $i+1$) are similar, for example,"SIGIR 2010"and "SIGIR 2011" are considered to be similar. For a particular conference $C_{v,i}$, the bias between its mth similar conference (Rm) and $C_{v,i+1}$ is calculated as:

$$bias(R_m, C_{v,i+1}) = sim(C_{v,i}, R_m) = \frac{sim(C_{v,i}, R_m) - sim(C_{v,i}, C_{v,i+1})}{sim(C_{v,i}, C_{v,i+1})}$$

We choose “SIGIR 1990” and “Telecommunication Systems 2003” from different research areas as examples shownin Figure 3-4. In total, 20 conferences are chosen to be evaluated as shown in Figure 5. From the plotting, the difference between the results by BSCS and $C_{v,i+1}$ almost all lie within 10% which proofs that BSCS algorithm is effective.

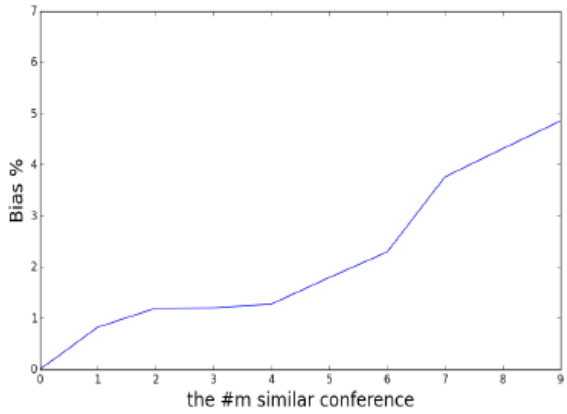

Figure 5. “Telecommunication Systems 2003”: Difference between Rm and “Telecommunication Systems 2004”

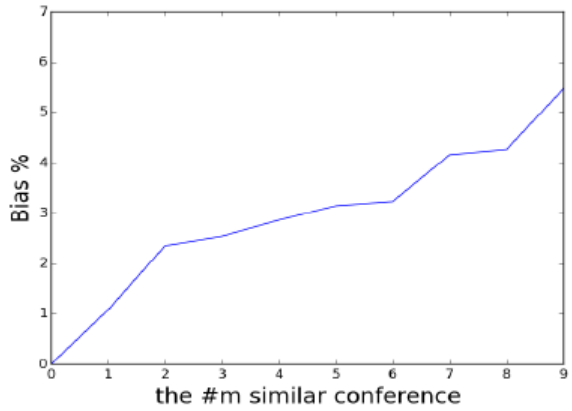

Figure 6. The Evaluation of 20 Conferences

- **Qualitative evaluation.**

Due to the space limitation, the results of 6 query conferences are shown in Table 2. From the conference names, we can observe that BSCS indeed can extract similar conferences. Taking “TKDD 2010” as an example, the conference names contains words like “computing”, “mining”, “learning”. More significantly relevant phrases such as “data mining”, “text mining” and “statistical analysis”, etc. appear in the results, highlighting the relevance between “TKDD 2010” and the results.

Query Conference	Top #	Similar Conferences
SIGIR 1990	1	Information Retrieval 1993
	2	Storage and Retrieval for Media Databases 2002
	3	Multimedia Information Systems 2002
	4	Adaptive Multimedia Retrieval 2007
	5	Information 2012
	6	Storage and Retrieval for Media Databases 2003
	7	Information Fusion 2010
	8	Information Fusion 2007
	9	Multimedia Information Retrieval 2004
	10	Belief Functions 2014
Telecommunication Systems 2003	1	IET Networks 2014
	2	Computer Networks and ISDN Systems 1992
	3	J. Sensor and Actuator Networks 2013
	4	Communications and Computer Networks 2005
	5	Neural Networks and Computational Intelligence 2003
	6	Ad Hoc Networks 2007
	7	Social Networks 2013
	8	NETWORKS 1993
	9	Social Networks 2010
	10	Journal of Communications and Networks 2011
Personal and Ubiquitous Computing 2008	1	J. Multimodal User Interfaces 2015
	2	User Modeling 2007
	3	Computer Aided Geometric Design 2011
	4	J. Computational Design and Engineering 2014
	5	User Interfaces for All 2002
	6	User Modeling 2005
	7	SIGMETRICS Performance Evaluation Review 2013
	8	J. Computational Design and Engineering 2015
	9	Personal and Ubiquitous Computing 2001
	10	User Modeling 2004
J. Network and Computer Applications 1999	1	Communications and Networking in Education 1999
	2	Designing Augmented Reality Environments 2000
	3	J. Computing in Higher Education 1992
	4	Interactive Learning Environments 2016
	5	EAI Endorsed Trans. Ubiquitous Environments 2015
	6	World Conference on Information Security Education 2003
	7	Advanced Programming Environments 1986
	8	Informatics in Higher Education 1997
	9	Building University Electronic Educational Environments 1999
	10	History of Computing in Education 2004
Graphical Models 2013	1	Object Representation in Computer Vision 1994
	2	Spatial Representation 2005
	3	ECAI Workshop on Knowledge Representation and Reasoning 1992
	4	Activity Context Representation 2011
	5	J. Visual Communication and Image Representation 2009
	6	Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation 2015
	7	Knowledge Representation and Organization in Machine Learning 1987
	8	Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning 2011
	9	Representation, Analysis and Visualization of Moving Objects 2010
	10	Knowledge Representation for Intelligent Music Processing 2009
TKDD 2010	1	BioData Mining 2008
	2	Active Mining 2003
	3	IADIS European Conf. Data Mining 2008
	4	Statistical Analysis and Data Mining 2010
	5	Context Sensitive Decision Support Systems 1998
	6	Data Structures and Efficient Algorithms 1992
	7	Statistical Analysis and Data Mining 2008
	8	BioData Mining 2012
	9	Community Computing and Support Systems 1998
	10	Ontologies and Text Mining for Life Sciences 2008

Figure 6. Results of 6 conferences

4. CONCLUSIONS

This paper proposes an algorithm that can help researchers find similar conferences. The proposed BFS-based framework is based on Lucence. Our proposed algorithm can efficiently retrieve relevant conferences compared with traditional methods. For evaluation, we employ DBLP dataset to measure the performance both quantitively and qualitatively.

REFERENCES

[1] G. Giannakopoulos, V. Karkaletsis, G. Vouros, and P. Stamatopoulos. Summarization system evaluationrevisited: N-gram graphs. ACM Trans. Speech Lang.Process., 5(3):5:1{5:39, Oct. 2008.

[2] S. Lawrence and C. L. Giles. Accessibility of information on the web. Intelligence, 11(1):32{39, Apr. 2000.

AUTHORS

I am currently a 4th year PhD student majored in Information System in Nanyang Technological University and SAP Innovation Center Network, Singapore. My research interests are recommender system, information retrieval and spatial-temporal data analysis.