Keyword Search in structured data and Network Analysis: a preliminary experiment over DBLP

Chiara Bernabei
and Francesco Guerra
Department of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
Email: {firstname.lastname}@unimore.it

Raquel Trillo-Lado
Department of Computer Science and Engineering
University of Zaragoza, Spain
Email: raqueltl@unizar.es

***Abstract*—Identifying similar items to the ones provided as input to a search system, is a challenging task. The main issues concern not only the management of large collections of data, but also the profiling of the users, who usually have different opinions, tastes and expertise.**

In this paper we make a preliminary investigation about the improvements in the accuracy of a search system provided by network analysis techniques supporting the discovery of relations among the items stored in the repository. For this reason, we have developed the SEEN prototype, a keyword search tool exploiting network analysis. SEEN has been evaluated against a relational version of the DBLP repository. The results of the preliminary experiments show that the the information provided by networks can improve the effectiveness of the results.

I. Introduction

Keyword-based search in large collection of structured data is indisputably a challenging task. First of all there is the strong need to develop techniques and tools providing real-time results. This goal is not always achieved by the existing systems [1] that can take hours for providing results. Moreover, the process for solving keyword queries has to intrinsically deal with loss of information. There is a triple semantic gap to bridge: firstly, the real word items are described according to a number of attributes and a vocabulary of values which "summarize and conceptualize" according to the database publisher'a perspective a possibly large set of features in a limited number of elements. The representation of items in a structured database can be itself cause of loss of information. Secondly, users formulating keyword queries have to select, among the possible terms describing the items they are looking for, the most representative to be included in the query, according to some criteria (e.g., the terms considered as the most relevant, important, specific, ...) and their vocabulary of known terms. The criteria adopted by users for selecting the keywords and the vocabulary owned determine the quality of the results retrieved. Thirdly, the techniques for solving keyword queries have to match and rank, based on some semantic measures, the users' keyword queries with the database data. Items can be conceptualized in different ways and using different vocabularies. This fact can hamper the retrieval of the data searched by the users. Moreover, several kinds of semantics and several kinds of measures can be adopted for matching and ranking the data associated to the keywords.

A wide range of techniques based on external knowledge (i.e., knowledge that is not extracted from the analysis of the datasource) have been proposed and evaluated in the literature for improving the accuracy of the results [2]. We divide these techniques in two main categories: techniques exploiting local knowledge (i.e., information related to the specific user formulating the query) and techniques exploiting global knowledge (i.e., details related to the entire database and to the users' plethora querying the database). Local knowledge can be exploited only if there is some mechanism implemented in the search engine able to identify and keep trace of the specific user formulating the query. This way, the users' behaviors can be profiled and the results retrieved by the search systems can take into account this knowledge for improving the accuracy (i.e., ranking the results according to the preferences established for users with a specific behavior). Nevertheless, the identification of the users can be difficult to achieve (this requires the management of accounts, and more computation power for solving the query). Thus making approaches based on local knowledge is not applicable in all the contexts.

Global knowledge techniques typically exploit ontologies, knowledge bases and lexical resources for interpreting and disambiguating the keywords (discovering and managing synonym, polysemic, hypernym/hyponym terms). Moreover, techniques based on the analysis of logs for discovering trends and frequent queries have been proposed.

In this paper, we claim that an additional global knowledge can be exploited in the process. This knowledge is generated by network analysis techniques that provide information about how the data in the source are related to each other and "where" a specific item is "located" in the network. Network analysis has been already coupled with keyword search [3]. Nevertheless only few systems deal with structured data sources (see Section II). We think that network analysis can be exploited for ranking the results (e.g., by ordering the results on the basis of their centrality in the network) or for explaining why a particular item is interesting for a user (e.g., the item is relevant because it belongs to a particular community). The idea can be furtherly extended by considering multiple networks built upon different attributes (or their combination), that describe the same data according to different criteria.

To evaluate the applicability and the effectiveness of our idea, we developed SEEN (SEarch with nEtwork aNalysis),

a preliminary implementation of a search system that couples keyword search with network analysis techniques for searching into bibliography descriptions extracted from the DBLP data source (http://dblp.uni-trier.de/). DBLP provides a basic description of the papers, where only information about title, authors and venue is recorded. We suppose not to have any information about users asking for papers. Inspired by [4], SEEN implements a "query by example" interface: users give a paper as a input, and the system retrieves the most similar papers in the repository. SEEN extends a typical keyword search engine, by supporting queries composed of keywords (e.g., the terms in a paper title) and other information (e.g., all the other data about a paper available in a database). SEEN exploits two types of global knowledge: the words contained in the titles and the authors of the paper. This knowledge supports the creation of two networks that reflect the relationships between authors (i.e., the co-authors' network) and words (e.g., the topic network), respectively. The co-authors' network, where nodes are papers and two papers are connected if they share an author, captures authors which tend to collaborate frequently with each other. The topic network, where papers are still nodes and two papers are connected if they share a word in the title, provides information about which words are used together. Our experiments shows that paper ratings improves by exploiting the knowledge generated by the combinations of the two networks.

The rest of these paper is structured as follows. Firstly, some related work is studied and analyzed in Section II. Then, the approach coupling keyword search and network analysis techniques is presented in Section III. After that, in Section IV the results of a set of experiments to evaluate our proposal are presented. Finally, some conclusions and future work lines are depicted in Section V.

II. Related Work

Nowadays, a great amount of information is stored in structured data sources, in particular in relational databases. So, easing the process of information search is a hot challenging task. Traditionally, information search on this kind of sources has been performed by means of formal query languages such as SQL. This kind of languages exploits information of the structure (metadata) and concrete values of the tuples, but final users have difficulties to deal with them. So, in the last decade new paradigms and interfaces such as keyword-based search on structured data sources or graphical interfaces have arisen to ease final users the process of information search [2], [5], [6].

Traditional techniques to search and recommend items are based either on similarity metrics that compare the values of the features of the items in the datasource with the values of the properties of the item provided as input (item-based recommendation systems) or on analyzing data about the tastes of the users or their interactions with the system of recommendation [7]. On the other hand, Network Analysis is a discipline that can be applied to every domain to extract knowledge (information hidden in the system). Thus, not only the general structure of the network is analyzed but also the properties of each element. In our case, information extracted is used to rank items provided as answer to a search process in a way that is consistent with the users' opinion. There are only few related works regarding this approach such as [4], [8] and [9]. As compared to the previous works, we aim at solving the problem of paper recommendation/search within a single data source by exploiting only a limited amount of information available in the data source. Similarly, we do not assume that users can be identified. So, no information about them is stored. The recommendation/search method proposed in this paper is also lightweight, in the sense that it does not require a heavy (pre-)processing, such as semantic extraction from the contents of the papers.

Other approaches based on "query by examples" or "exemplar queries" have been proposed in [10]. That work focuses on the specification of the semantics of such queries and show that they are different from other approaches such as approximate and related queries. Moreover, it also provides us with an implementation of those semantics for graph-based data. In contrast, our proposal considers network analysis to obtain and rank the items provided to answer to a particular query, i. e. the graphs (or networks) considered in the process are built from the information stored in the extension of the data-source.

III. Combining keyword search and network analysis techniques

SEEN requires a paper as input and retrieves a ranked list of papers with similar characteristics. The selection and ranking of the papers to be associated with the users' inputs is performed according to specific features. The selection of the ones to be adopted is a critical task, since they are domain and (in most of the cases) user dependent. In this preliminary proposal, we decided to use two kinds of features: domain specific features (also called baseline features) and the network analysis generated features. Domain specific features describe the paper as it is. They provide simple characteristics associated to the papers in isolation (year, venue, topics, ...). Network Analysis features describe how papers are related with each other. In this implementation, we have focused on two kinds on information: the paper titles and authors. This choice is justified by two main reasons: firstly, the limited information contained in the database: DBLP contains only basic information about the papers. The second reason is the simplicity: the goal of the application prototype is actually to evaluate how the accuracy of a keyword search system can be improved by the application of network analysis techniques.

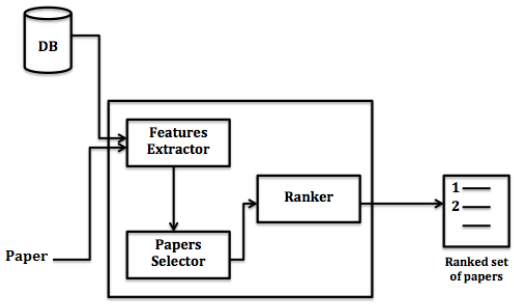

Fig. 1. The SEEN functional architecture.

A. System modules

Figure 1 shows the SEEN functional architecture which is composed of three main components: the feature extractor, the paper selector and the ranker.

As data source, we adopted a relational implementation of DBLP database adopted. In particular, we focused on the tables collecting data about publications (*article*, *book*, *collection* and *in_proceeding*), containing details about the authors, and describing the relationships among authors and their publications.

1) Features Extractor: The feature extractor module is responsible of the creation of a matrix (see Figure 2) describing the papers in the database[1]. The rows in the matrix represent the papers which are described by the features represented in the columns. The number of rows corresponds to the number of papers (more than 1M), and the number of columns depends on the number of features adopted. The ones used in SEEN can be divided into two different sets: baseline features and network features.

Baseline features describe the main features of the papers in isolation. In SEEN, we used the most important words appearing in the titles. We tried some heuristics for deciding which are the most important words: after a process of stemming and stop-words deletion, we experiments the selection of the top n most frequent words appearing in the titles of the papers. As an alternative, we tried the selection of the top n words which maximize some centrality measures in the topic network. n is a system parameter: the higher n, the more accurate result will be computed. Nevertheless, the selection of a high n value increases the execution time. The values of the entries in the matrix are computed by exploiting classic Information Retrieval (IR) models. In particular, we adopted a Boolean IR model, where each element in the matrix indicates whether the paper considered (the row of the matrix) contains (value 1) or does not contain (value 0) the keyword (the term) corresponding to the column of the matrix. As an alternative, we can adopt an Absolute-frequency or a TF_IDF model for populating the matrix. The first model is inspired by the earlier Vector Space IR models, where the elements in the matrix indicate how many times the keyword corresponding to the column of the matrix appears in the title of the considered paper. The second is inspired by more modern Vector Space IR models, where the length of the documents and the content of the corpus analyzed are considered. Thus, in this case, the length of the titles and the vocabulary of the set of title of the database are considered to define the value of each element of the matrix.

Network analysis features provide a "global" representation (i.e., a representation which takes into account the other papers in the database) of the paper. We used six features extracted from the networks analyzed: three come from co-authors' network, others three from the topic network. The features extracted from the co-authors' network exploited in SEEN are [11]:

- Sum of the degree of the authors writing the paper. The degree measure the number of connections that a node has with others. In social networks nodes with highest degree tend to have a greater ability to influence other nodes. Consequently, authors with high degree values are the most influencers and papers with highest degree are the ones written by the most important authors in the network;
- Sum of the betweenness of the authors writing the paper. The betweenness measures the number of shortest paths passing through a node. Therefore it detects nodes that connect different groups in the network and carry the highest volume of traffic. The authors with high betweenness values are central with respect to the network;
- Sum of the PageRank of the authors writing the paper. This measure represents how important is a node depending on the importance of its neighbours and how an author collaborates productively with his colleagues.

Analogous features are extracted from the topic network:

- Sum of the degree of the words in the title. The degree gives a measure of the the relevance of the words used in the paper titles;
- Sum of the betweenness of the words in the title. This measure report how important is a word in connecting groups of words. Words with high betweenness values are adopted in different contexts;
- Sum of the PageRank of the words in the title, which expresses the importance of the words depending on the importance of the words used in the titles of the papers;

Finally, the values of the features have been normalized, as usually happens in IR, not to have heterogeneous attributes.

2) Papers Selector: This component retrieves the paper similar to the ones provided in input. First of all the features of the input paper have to be retrieved. If the paper is already in the database, this is a simple task, performed by a simply query on the database. Otherwise, a vector representing the baseline and the network features has to be created. For retrieving similar papers, we adopted a Locality-Sensitive Hashing (LSH) based approach.

LSH is a method of performing probabilistic dimension reduction, i.e. reducing the number of variables taken into consideration, of high-dimensional data, given an approximation of the exact nearest neighbor. The basic idea is to hash items several times, in such a way that similar items are more likely to be hashed to the same bucket than dissimilar items are; then only pairs that hashed to the same bucket are considered as candidate pairs and check for similarity. The hope is that most of the dissimilar pairs will never hash to the same bucket, and therefore will never be checked. False positive and false negative can occur, but it is desirable that there is only a small fraction of the total [12].

In our implementation, the cosine distance has been selected for identifying similar papers. In particular, the baseline features of the input paper are compared, by using LSH, with the baseline features of all papers in the database. The number

[1]Only papers from 2008 to 2013 are used in the experiments for simplicity and performance reasons.

Title	Baseline Features				Networks Features					
	Word 1	Word 2	··· ···	Word n	Degree of authors	Betweenness of authors	PageRank of authors	Degree of words	Betweenness of words	PageRank of words
Semantic Heterogeneity Issues on the Web.	0	0	···	0	0.041	0.003	0.046	0.263	0.146	0.258
Feature Selection in Text Mining.	0	0	···	0	0.039	0.013	0.048	0.0272	0.086	0.225
Keyword search over relational databases: a metadata approach.	0	0	···	0	0.106	0.010	0.093	0.403	0.208	0.365
A system for network analysis.	1	1	···	0	0.15	$5.2{\cdot}10^{-9}$	0.06	0.77	0.90	0.84
...	...	...	...	...	...	...	...	...	...	...

Fig. 2. Example of the matrix used.

of baseline features is an important parameter that limits the success of the system. In fact, if a word appearing in a title is not defined as a baseline feature, this cannot be used for selecting the paper. This is a critical task, the process for selecting the papers returns papers that will be the input of the Ranker component: if the result of the selection is not accurate, also the second step will fail. Note that the number of papers returned as output can vary: we performed this choice through a parameter that we set to 20, thus returning to the users only the top 20 results. A lower value must be used if we want to consider only papers with titles very close to the one in input; a higher value must be used if we are also interested in papers without all the words belonging to the title of the input paper.

3) Ranker: The Ranker module ranks the results retrieved in the previous step. The network analysis results allow us to provide several kinds of ranking giving more relevance to some features according to the users' preferences.

We implemented two ways for performing this task. Firstly, we considered the networks features and we ranked the papers which maximize these values. This way, the results provided will be the ones with the "best" authors and the "best" words according to the measures adopted in the analysis. Specific customized ranks, by weighting some features more than others, can be easily provided. If, for example, users are interested in papers with the most important authors, we can weight according to some coefficient the corresponding features. The second possibility is to rank the results according to how much their network features are close to the one of the input paper. In this way, we provide high rank to papers having similar "importance" values to the one in input.

IV. Experimental Evaluation

The goal of the experimental evaluation is to demonstrate the effectiveness of the approach proposed, i.e., if network analysis actually improves ranking. We do not consider the efficiency: SEEN takes 350 seconds in average to compute the similarity. Clearly, reducing the computation time and optimizing the execution is definitely a future work if we want to make the system usable in real environments.

A. Systems and metrics used for the evaluation

To compare the results provided by our proposal with other state-of-the-art techniques, other two systems have been used as a reference: Google Scholar and Indri. Google Scholar (https://scholar.google.it/) is a web search engine that indexes the academic literature. The way in which Google Scholar ranks the papers is unknown and subject to industrial patents. Indri is a module of the Lemur Project (http://www.lemurproject.org), an open-source framework software, used for developing search engines, text analysis tools, browser toolbars, and data resources in the area of IR. The model adopted in Indri for retrieval is described in [3].

SEEN has been evaluated using two metrics:

1) Precision: it is defined as the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved.

2) Correlation [13]: we adopted the Spearman correlation coefficient [14] to describe the strength of a linear relationship between two variables [15], i.e., the rankings. It returns a value between -1 and 1, where 1 is the perfect agreement between the rankings, i.e., the two rankings are the same, 0 means that the rankings are completely independent, and -1 is the perfect disagreement between the rankings, i.e., one ranking is the reverse of the other. Spearman's

correlation coefficient is defined as:

$$\rho = 1 - \frac{6 \cdot \sum_i D_i^2}{N \cdot (N^2 - 1)}$$

where D_i is the difference between the ranking in the position i, and N is the total number of observations. Spearman's correlation takes into account both the number of concordant pairs and the distance between two rankings. Therefore, this measure considers as correlated rankings with "similar" common positions.

B. Methodology

The evaluation of the system has been done using three different methods, as described in Sections IV-B1, IV-B2, IV-B3. In this preliminary proposal, we evaluated the performance obtained with 10 papers.

1) Evaluation measuring precision: The precision is computed by comparing the results of our system with the ones provided by Google Scholar and Indri. All the systems have been queried with the same queries (10 papers) and the top-20 results returned have been taken into account for the evaluation of the precision.

	Precision Google	**Precision Indri**
Paper 1	25%	65%
Paper 2	5%	15%
Paper 3	5%	5%
Paper 4	100%	10%
Paper 5	15%	0%
Paper 6	5%	5%
Paper 7	5%	5%
Paper 8	5%	10%
Paper 9	5%	5%
Paper 10	5%	10%

TABLE I. PERCENTAGE OF PRECISION FOR EACH PAPER. THE COMPARISON IS DONE WITH BOTH GOOGLE SCHOLAR AND INDRI.

As shown in Table I, the values of precision are in general not very high, except for Paper 1. The detailed analysis has pointed out that even if SEEN does not return many papers in common with Indri and Google Scholar, most of the results are consistent with the query proposed, and then they can be considered as acceptable. Moreover, among the results retrieved, in most of the cases there are several papers with the same (or close) similarity level. Since we limited the analysis to 20 papers, it happened that we cut the results to a number of papers excluding other with a similarity level equal to the ones in the result set. Finally, a critical aspect of our system is the number of words chosen as baseline features. We set this parameter to 1500: not all words appearing in the titles are included. There are some cases in which the absence of a word (e.g. *RDF*) drastically modifies the result, i.e., the returned titles by SEEN are not as consistent as those provided by the other systems.

2) Evaluation according to the number of citations: We compared the top-20 results provided by our system and Indri taking into account the number of citations. The rationale is that the number of citations can be considered as a rough measure of the paper importances: the more the number of citations, the more the paper is good and deserves to be ranked in the highest positions in the result set. We introduced a 2-step procedure for evaluating the systems:

1) A reference list of results is created starting from the one generated by SEEN and Indri. The score (and consequently the order) attributed to each paper p_i in the result is computed according to the number of citations retrieved in Google Scholar (ci), normalized with respect to the year, by adopting the following formula:
$$\frac{4}{CY - YP} \cdot ci$$
where CY is the current year, YP is the year of publication of the paper taken into account and ci is the number of citations for that paper;
2) The Spearman's correlation between the actual result set provided by SEEN and Indri, and the respective reference one is computed. We tested two possible ranks for the actual result sets: in the first rank, the papers are ordered taking into account all the values of network features (i.e., the importance-based result set); in the the second rank, we ordered the results according to the their similarity with the paper given in input (i.e., the similarity-based result set)).

Table II shows the values of correlation for each paper. Considering the importance-based result set, the correlation is positive for half the paper and negative for the other half. For similarity-based result set there are more papers correlated positively that those correlated negatively. Note that, since we have more papers with the same score (their order does not depend on their quality), values close to zero can be considered as strong correlations.

	Importance result set (SEEN)	**Similarity result set (SEEN)**	**Importance result set (Indri)**	**Similarity result set (Indri)**
Paper 1	-0.40	+0.50	-0.20	+0.20
Paper 2	-0.18	+0.14	-0.25	-0.11
Paper 3	0.04	-0.20	-0.06	+0.35
Paper 4	+0.13	-0.02	+0.23	-0.01
Paper 5	-0.15	-0.21	+0.01	+0.14
Paper 6	-0.05	+0.31	+0.004	+0.07
Paper 7	+0.15	0.07	+0.23	-0.11
Paper 8	+0.13	0.05	0	0.64
Paper 9	-0.06	-0.03	-0.17	0.15
Paper 10	-0.38	+0.46	+0.33	+0.40

TABLE II. CORRELATION BETWEEN THE SYSTEM RANKINGS AND THE NUMBER OF CITATIONS.

3) Evaluation by means of human testers: In this test, we evaluated if the results provided by SEEN are relevant for the users. We provided the testers with three different files containing the same result set obtained by 5 queries. The files differ since three different types of information are shown:

1) list T: for each query, the paper title in input, and the top-10 titles retrieved by SEEN and randomly ordered are provided;
2) list TA: for each query, the paper title and authors in input, and top-10 titles (with authors) retrieved by SEEN and randomly ordered are provided;
3) list TAC: for each query, the paper title, authors, year of publication and conference in input, and the top-10 titles retrieved with authors, year of publication and conference (in random order) are provided.

The testers are asked to rank the papers in the result sets (following the order T, TA and finally TAC). So, by ranking T,

they are not influenced by the information provided in TA and TAC, and by ranking TA they are not influenced by additional data given in TAC.

	T	TA	TAC
Paper 1	0.34 (0.28)	0.35 (0.21)	0.32 (0.3)
Paper 2	0.44 (0.31)	0.4 (0.25)	0.45 (0.21)
Paper 5	0.43 (0.33)	0.29 (0.31)	0.17 (0.32)
Paper 8	0.61 (0.21)	0.53 (0.23)	0.51 (0.21)
Paper 10	0.39 (0.26)	0.3 (0.3)	0.29 (0.26)

TABLE III. AVERAGE CORRELATION BETWEEN USERS' RANKINGS. VALUES IN BRACKETS REPRESENT THE STANDARD DEVIATION.

The results are then analyzed according to two perspectives. Firstly, for each paper the correlation between the rankings provided by the users with each other is computed. This way, we discover the heterogeneity of the users' opinions and how the opinions change on the basis of the available information. The results of this analysis are shown in Table III, where the average and standard deviation of the Spearman's correlation coefficient is reported. The results show that users' opinions agree with each other. The correlations among users are almost always positive, and the averages are always positive. Moreover, the table shows that users base their rankings mainly on the words appearing in the titles. The correlation among users rankings is lower with the increase of information: this means that the set of testers selected for the evaluations could not be really expert in the topic.

In the second experiment, we measured the Spearman's correlation coefficient computed between the ranking provided by the users and the ranking generated by SEEN. The results are summarized in Table IV-B3, where the average correlation is reported. Column TAC shows the results obtained in normal usage conditions: three of five papers tested show positive correlation. In these cases, the system is able to provide a useful ranking that reflects the users' opinions.

	T	TA	TAC
Paper 1	-0.04	0.008	0.03
Paper 2	0.13	0.10	0.11
Paper 5	-0.39	-0.37	-0.25
Paper 8	0.54	0.55	0.55
Paper 10	-0.22	-0.17	-0.13

TABLE IV. AVERAGE CORRELATIONS BETWEEN USERS' RANKINGS AND OUR SYSTEM FOR EACH PAPER AND LIST PROPOSED.

V. CONCLUSIONS AND FUTURE WORK

The SEEN tool exploiting network analysis to find and ranking similar items to the ones provided is proposed and evaluated.

The advantage of the proposal is that it does not require either personal information of the users or data about their interactions with the system to provide effective results. So, SEEN can be easily incorporated as a module into search systems for structured datasources.

Future work will be devoted to improve the experimentations, by considering more users and more queries and the techniques implemented. Moreover we plan to exploit network analysis techniques to provide an explanation of the relevance of the papers in the result set. In particular, we can say that a specific paper is relevant since it has been written by the "best" authors in the network, or is relevant since it is multi-disciplinary because of authors and words used have high betweenness values. We think that similar explanations of the results could improve the effectiveness of the results.

ACKNOWLEDGEMENT

This article is based upon work from COST Action KEYSTONE IC1302 (www.keystone-cost.eu), supported by COST (European Cooperation in Science and Technology).

REFERENCES

[1] J. Coffman and A. C. Weaver, "An empirical performance evaluation of relational keyword search techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 30–42, 2014.

[2] J. X. Yu, L. Qin, and L. Chang, *Keyword Search in Databases*, ser. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2009. [Online]. Available: http://dx.doi.org/10.2200/S00231ED1V01Y200912DTM001

[3] D. Metzler and W. B. Croft, "Combining the language model and inference network approaches to retrieval," *Inf. Process. Manage.*, vol. 40, no. 5, pp. 735–750, 2004. [Online]. Available: http://dx.doi.org/10.1016/j.ipm.2004.05.001

[4] C.-Y. Teng, Y.-R. Lin, and L. A. Adamic, "Recipe recommendation using ingredient networks," in *Proceedings of the 4th Annual ACM Web Science Conference*, ser. WebSci '12. New York, NY, USA: ACM, 2012, pp. 298–307. [Online]. Available: http://doi.acm.org/10.1145/2380718.2380757

[5] S. Bergamaschi, E. Domnori, F. Guerra, R. T. Lado, and Y. Velegrakis, "Keyword search over relational databases: a metadata approach," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011*, 2011, pp. 565–576. [Online]. Available: http://doi.acm.org/10.1145/1989323.1989383

[6] S. Bergamaschi, F. Guerra, M. Interlandi, R. T. Lado, and Y. Velegrakis, "QUEST: A keyword search system for relational data based on semantic and machine learning techniques," *PVLDB*, vol. 6, no. 12, pp. 1222–1225, 2013. [Online]. Available: http://www.vldb.org/pvldb/vol6/p1222-guerra.pdf

[7] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems*. Cambridge University Press, 2010, cambridge Books Online. [Online]. Available: http://dx.doi.org/10.1017/CBO9780511763113

[8] D. Steidl, B. Hummel, and E. Jrgens, "Using network analysis for recommendation of central software classes." in *WCRE*. IEEE Computer Society, 2012, pp. 93–102. [Online]. Available: http://dblp.uni-trier.de/db/conf/wcre/wcre2012.htmlSteidlHJ12

[9] X. Li and L. Chen, "Recommendations based on network analysis," in *Advanced Computer Science and Information System (ICACSIS), 2011 International Conference on*. IEEE, 2011, pp. 9–16.

[10] D. Mottin, M. Lissandrini, Y. Velegrakis, and T. Palpanas, "Searching with XQ: the exemplar query search engine," in *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, 2014, pp. 901–904. [Online]. Available: http://doi.acm.org/10.1145/2588555.2594529

[11] E. Yan and Y. Ding, "Applying centrality measures to impact analysis: A coauthorship network analysis," *CoRR*, vol. abs/1012.4862, 2010. [Online]. Available: http://arxiv.org/abs/1012.4862

[12] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. New York, NY, USA: Cambridge University Press, 2011.

[13] J. Coffman and A. C. Weaver, "Learning to rank results in relational keyword search." in *CIKM*, C. Macdonald, I. Ounis, and I. Ruthven, Eds. ACM, 2011, pp. 1689–1698. [Online]. Available: http://dblp.uni-trier.de/db/conf/cikm/cikm2011.htmlCoffmanW11

[14] G. W. Corder and D. I. Foreman, *Nonparametric Statistics: A Step-by-Step Approach, 2nd Edition*. Wiley, 2014. [Online]. Available: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1118840313.html

[15] D. Diez, C. Barr, and M. Çetinkaya-Rundel, *OpenIntro Statistics: Second Edition*. CreateSpace Independent Publishing Platform, 2012. [Online]. Available: http://books.google.it/books?id=OtyCMAEACAAJ