International Journal of Intelligence Science, 2022, 12, 9-20
https://www.scirp.org/journal/ijis
ISSN Online: 2163-0356
ISSN Print: 2163-0283

Recommender System for Information Retrieval Using Natural Language Querying Interface Based in Bibliographic Research for Naïve Users

Mohamed Chakraoui[1], Abderrafiaa Elkalay[2], Naoual Mouhni[3]

[1]LS3M, Polydisciplinary Faculty of Khouribga, University Soultan Moulay Slimane, Béni Mellal, Morocco
[2]RITM, EST University Hassan II Casablanca, Casablanca, Morocco
[3]LAMIGEP EMSI Marrakech, Marrakech, Morocco
Email: chakraoui@gmail.com, elkalay@hotmail.fr, na.mouhni@gmail.com

How to cite this paper: Chakraoui, M., Elkalay, A. and Mouhni, N. (2022) Recommender System for Information Retrieval Using Natural Language Querying Interface Based in Bibliographic Research for Naïve Users. *International Journal of Intelligence Science*, **12**, 9-20.
https://doi.org/10.4236/ijis.2022.121002

Received: October 17, 2021
Accepted: January 24, 2022
Published: January 27, 2022

Abstract

With the increasing of data on the internet, data analysis has become inescapable to gain time and efficiency, especially in bibliographic information retrieval systems. We can estimate the number of actual scientific journals points to around 40,000 with about four million articles published each year. Machine learning and deep learning applied to recommender systems had become unavoidable whether in industry or in research. In this current, we propose an optimized interface for bibliographic information retrieval as a running example, which allows different kind of researchers to find their needs following some relevant criteria through natural language understanding. Papers indexed in Web of Science and Scopus are in high demand. Natural language including text and linguistic-based techniques, such as tokenization, named entity recognition, syntactic and semantic analysis, are used to express natural language queries. Our Interface uses association rules to find more related papers for recommendation. Spanning trees are challenged to optimize the search process of the system.

Keywords

Recommender Systems, Collaborative Filtering, Apriori Algorithm, Natural Language Understanding, Bibliographic Research

1. Introduction

Due to the huge volume of information on the internet, data are now the most valuable thing. Still, is it easy to have useful information in a short time as well as

the quantity required? The answer is obviously "No". The information is everywhere and our access to information far exceeds our ability to capture and classify it for use. We need to seek technological means for the automatic summary of information and its classification according to our needs because the flow of information on the internet is always increasing.

"Text summarization" may be defined as the technique to create a short and accurate summary of longer text documents. Automatic text summarization will help us with relevant information in less time. Natural language processing (NLP) plays an important role in developing an automatic text summarization.

With the expansion of machine learning, deep learning and big data, recommender systems have gone beyond their existence in the areas of consumption and markets. Scientific Research is a noble field that has also benefited from recommendation systems in the past few years. A primary application of natural language processing (NLP) is answering questions. Search engines like GOOGLE puts the world's information at their fingertips, but search engines cannot make answers to questions asked by humans. Managing the information is a challenge because of the widely usage and progressive development of computer systems or software to exchange information from/to different databases. Information retrieval from various sources has become a hot topic in the last few years [1].

As in all fields, Scientists and Researchers recognize the ranking of international journals. Organizations such as Web of Science, Scopus, DBLP, IEEE and others came to make this classification. Naïve users cannot gather their needed articles from Scopus and Web of Science in the same time and make recommendations through Search Engine. The journal impact factor given by recognition organizations such as Web of Science and GIF is very important to search papers and get results following their degree of relevance of the criteria.

The proposed Interface interprets bibliographic queries expressed in controlled natural language and returns relevant bibliographic papers following the order of relevance and impact factor. Natural language queries supported in this work are used to express complex nominal phrases that describe bibliographic entities. We make easy query interpretation, processing and visualization in different bibliographic domains. We describe a practical study of data mining tool on bibliographic research based on association rule mining and natural language processing. The main objective is to make a mining tool in which the optimal information can be found easily following many predicates written in a natural language. Currently, a lot of general tools and data mining frameworks are available to the end user. Some of them are discussed by [2].

The paper is organized as follows: Section 2 describes the related works about natural language processing, information retrieval and association rules in the last few years; Section 3 is about the proposed mining tool and some examples of association rules used in this current. Finally, conclusion and perspectives are outlined in Section 4.

2. Related Works

2.1. Database Systems

The most solicited issues in database fields are efficiency, speed and reliability of relational database systems. Many papers have discussed these topics as well as object oriented databases and other databases architecture, but these last remain insufficient and could not get to the top requested by researcher community [3]. Our paper comes in this context to a progressive thread with a recommender system to information retrieval, provides a complete theme, supported by experiments, yielded results, and then make specialists in this interesting discussion.

Indexing is the most suitable way to optimize database systems. Furthermore, parallelization is one of the top ways of optimizing index [4]. The purpose is to speed the data processing and decrease the response time of complex queries. To achieve that, query optimization remains among the best solution.

A poorly written query can increase the input-output gets, which leads to increase in the execution time. Then slow down the system. To solve this problem, we proposed a semantic and syntactic corrector based on the English language.

When a request is sent to the RDBMS (Relational Database Management System), it will be parsed and translated into RDBMS language, then the RDBMS establishes several executions plans possible, then the RDBMS optimizer chooses the most suitable one and runs it [5]. In this background, the system is connected to two different databases that we have previously configured via API: Web of Science and Scopus. For each end-user, the request is made independently of the database but the system brings articles indexed in web of science followed by scopus... This system is described through different sections of this paper.

2.2. Information Retrieval

Information retrieval is the process of identifying items containing information relevant to a given query. It is useful to proof, organize and store a set of data. However, the source of information (databases) is widely different; end-user can access the sought information very easily. The source of information can be structured (relational databases); semi-structured (XML, LATEX, Scientific data…) or unstructured data (text documents, Email messages, Audio files…). Information retrieval informs the end-user to get the result of the sought data and its source and the number of founded data [1].

2.3. Recommender Systems

Recommender systems aim to provide personalized recommendations to users for specific items (e.g., music, books, movies,). Popular techniques involve content-based models and collaborative filtering approaches [6]. In Content-Based Filtering Systems, a user profile represents the content descriptions of items to which that user has previously expressed interest. The content descriptions of items are represented by a set of features or attributes that characterize that item. The recommendation generation task in such systems usually involves simili-

tude-extracted features from unseen or unrated items with content descriptions in the user profile. Items considered sufficiently similar to the user profile are recommended to the user [7]. Collaborative Filtering is the nearest neighbor method. Given some user profiles, it predicts whether a user might be interested in a certain item, based on a section of other users or items in the database. Traditionally, the primary technique used to accomplish this task is the algorithm of standard memory-based k-Nearest-Neighbor (kNN) classification approach which compares a target user's profile with the historical profiles of other users to find the top k users who have similar tastes or interests [7]. There are in general two types of collaborative filtering: user-based and item-based. Often, they share the same concept but they vary in how the neighborhood is formed [8]. In user-based, collaborative filtering, recommendations are generated by considering solely the ratings of users on items, by computing the pairwise similarities between users. In Item-based collaborative filtering, the similarities are computed between each pair of items. They currently recommend papers based on coupling of item-based technique and content-based filtering approach, which can give large relevant items to the end-user.

2.4. Association Rules

Association rule was a tool which help vendors to find a set of items that are commonly accessed or purchased together. By means of association rules, we can organize web sites, vitrines and supermarkets to put more common content closely. As an example, cross-sale product recommendations is one of the effective means for supermarkets to buy as many items as possible, [7].

Association rule discovery techniques, such as Apriori algorithm [9], were initially developed as a set of techniques for mining supermarket basket data analysis but have since been used in various domains including social networks, medical analysis and others [9] [10]. Apriori is an algorithm for frequent mining of a set of items in transactional databases and learning of related rules. It identifies the individual items that occur frequently in the database and extends them into a larger set of items as long as they occur frequently in the database. Apriori identifies frequent item set to be used to determine association rules that highlight common trends in the database [11].

2.5. Spanning Trees

Graphs are structures which map relations between objects. The objects are called "nodes" and the connections between them are named "edges". Edges and nodes are commonly referred to by several names that mean exactly the same reference. The purpose of using graph manipulation and analysis (python NetworkX) is that concepts and terminology are generally intuitive. Given a connected and undirected graph, a spanning tree of that graph is a subgraph, which forms a tree and connects all the vertices together. A single graph can have many different spanning trees. A minimum spanning tree (MST) or minimum weight

spanning tree for a weighted, connected and undirected graph is a spanning tree with weight less than or equal to the weight of every other spanning tree. The weight of a spanning tree is the sum of weights given to each edge of the spanning tree. The principal feature of this graph is related to the fact that the vertices of the graph are partitioned into a certain number of clusters [12]. **Figure 1** shows an explanation of the clusters situation.

MST based clustering algorithm is employed with Kruskral algorithm [13]. This allows us to set a threshold value and step size. Edges with lengths which are greater than the threshold value, are removed from the MST. We then calculate the ratio between the intra-cluster distance and inter-cluster distance and record the ratio as well as the threshold. We update the threshold value by incrementing the step size. Every time we obtain the new (updated) threshold value, we repeat the above procedure. We stop repeating, when we encounter a situation, in which the threshold value is maximum. In such situation, all the data points belong to a single cluster. Now, we obtain the minimum value of the recorded ratio and form clusters corresponding to the stored threshold value. To benefit from multicore computer, we parallelize this algorithm using the "distributed memory architecture". Filter-Kruskal algorithm avoids sorting edges which are obviously not in the MST [14].

2.6. Natural Language Understanding

Natural language interfaces (NLI) are used to query structured information stored in databases. There are many types of NLI, the most solicited one is natural language interfaces to databases (NLIDB), in which a relational database is used to store structured information [15]. The disadvantage of this type is that of SQL is too difficult for most non-computer scientist users. Then, we need a representation that is both "human understandable" and "RDBMS understandable" [16]. Another type of NLI is natural language interfaces to knowledge bases (NLIKB) that use an ontology to manage information [15]. Natural language Interfaces can communicate with all RDBMS, and use close algorithms for interpretations of NLI queries and mapping them to RDBMS queries. Queries in database systems are optimized using local parallel index partitioning.

2.7. Pilot Study

Initially, we needed to find out how researchers would interact with the system

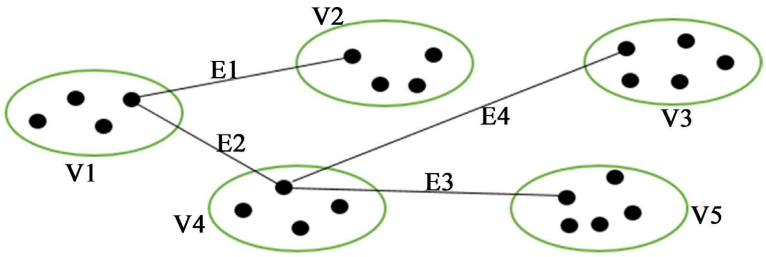

Figure 1. An instance of our clusters MST.

in natural language. For this purpose, we recruited subjects and asked them to describe how to solve problems with the aid of search engines. Google scholar, Scopus, Thomson routers, and others were provided together with these tasks. The main idea is to imagine explaining to a human partner how to solve a given task and to write down what they would say. Even though there were no suggestions to ask the imaginary partner for help, many participants did just that.

Many works have been proposed to discuss the set of recommender systems, databases analysis and machine learning together. Most of these rely on pairwise association rules [2].

The cold start problem that is related to recommendations for novel users or new items [6] will be also treated in this current. Hybrid recommender system based on knowledge and social networks [17] will not be discussed in this paper. According to [18] recommender systems can be partitioned to four different approaches as follows:

- Content-based recommender systems: they try to find products, services or contents that are similar to those already evaluated by the user. In this kind of systems, user's feedback (that can be collected in many ways) are essential to support and accomplish recommendations.
- Knowledge-based recommender systems: they model the user profile in order to, through inference algorithms; identify the correlation between their preferences and existing products, services or content.
- Collaborative filtering recommender systems: they create/classify groups of users that share similar profiles/behaviors in order to recommend products, services or content that has been well evaluated by the group to which a user belongs.
- Hybrid recommender systems: they combine two or more techniques to improve the "quality" of recommendations.

3. The Experimental Study

Naïve user inputs a natural language query through natural language Interface that is translated to an SQL query using the following phases:

- Stop word removal.
- Stemming.
- Content word extraction.
- Syntactic Analysis.
- Candidate query formulation.

The first phase consists of taking off stop words using a predefined list in toolkit. The second phase consists of processing the root word extraction for the other words. The meaningful words are extracted using syntactic parsing phase that consists of a top-down parser [19].

Overall, there were 50 naïve users. In a self-assessment, 15% evaluated as experts, 65% advanced users, and 20% beginners with regard to search engines and computer using. The participants' experience on working with search engine was

on average 14.5 years, with a maximum of 20 years and a minimum of 7 years. This current discusses a running example from the empirical study that involves a natural language interface for information retrieval system programmed with Python 3.6 connecting with two different databases via APIs to execute our methods easily as illustrated in Figure 2.

The interactive system should help inexperienced users by asking for missing relevant information (auto completion) because human natural language input can contain gaps. Nevertheless, natural language has the ability to resolve references of already provided information (history management), and successfully resolve ambiguity in natural language input (autocorrect).

Figure 3 shows that the system is not only successful in resolving references to the previously provided information, but also acts actively and asks relevant questions depending on the dialog context as the following:

Our proposed recommender system built on item based technique and content-based filtering approach recommendations system can act like the Figure 4:

Your domain of research: machine learning, natural language processing, recommender system…

The recommendations are listed as following.

4. Results and Findings

This section validates the developed application in this current. To evaluate the

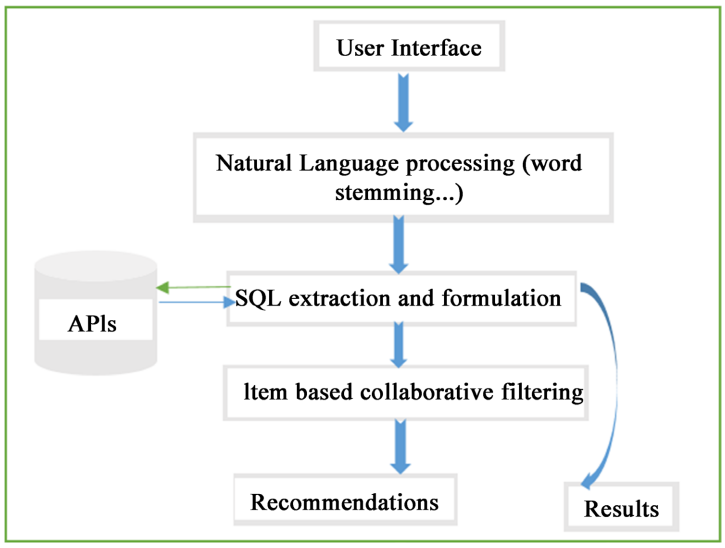

Figure 2. Architecture of the system.

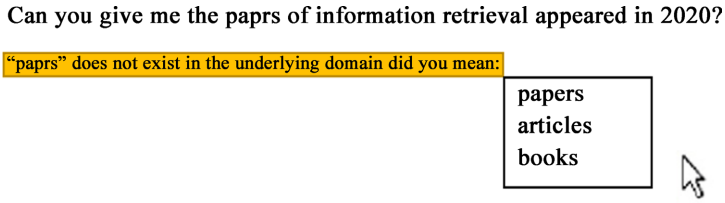

Figure 3. Syntactic and semantic completion.

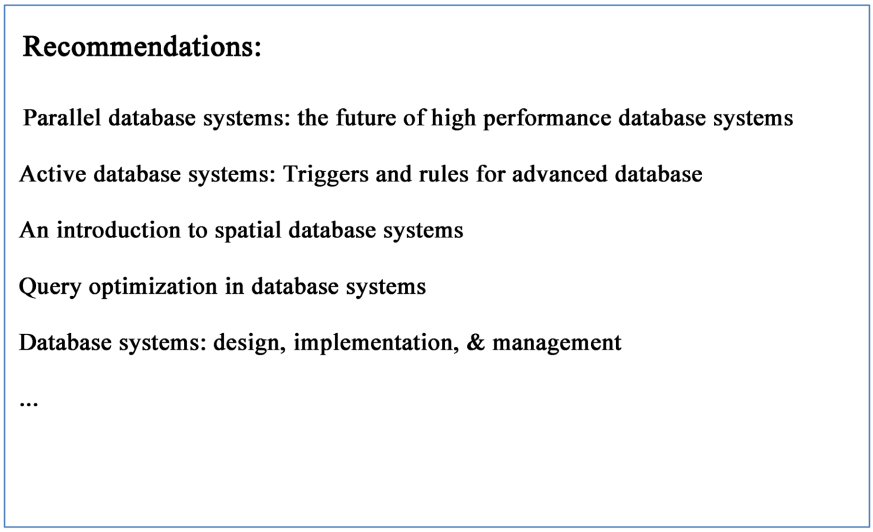

Figure 4. Recommendations interface based on domain interest and request.

quality of the bibliographic recommendations made by this system, we used precision and recall as metrics.

Precision measures the percentage of the content results, which are relevant for users. It can be calculated by the following formula:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4}$$

Recall refers to the percentage of total relevant results correctly classified by our system. It can be calculated by the following formula:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{5}$$

We can also calculate accuracy to know immediately whether our system is being trained correctly and how it may perform generally. It can be calculated by the formula:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total}} \tag{6}$$

where:

True positive = number of queries understood and properly processed by the system.

True negative = number of queries understood and not properly processed by the system.

False positive = number of queries not understood and properly processed by the system.

False negative = number of queries not understood and not properly processed by the system.

It is a very common situation where we end up with a model where either Precision is high and Recall is low or vice versa. It becomes a little difficult with their two metrics to evaluate our model and say which is better. It would be a lot

easier if we had a single value to measure performance, and that metric is F1 score. F1 score is defined as the harmonic means of Precision and Recall (because the general average does not penalize the extreme values). F1 Score:

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

Indeed, this concept is widely applied to information retrieval systems.

The interactive system should help inexperienced users by asking for missing information because natural language input may be inputted wrongly, have the ability to resolve keyboarded references, and successfully resolve ambiguity in natural language input [20].

Ambiguous information is among the disadvantages of natural languages. This current deal with ambiguous queries by adding intermediate dialog box for clarification. Each one can clarify the semantic meaning until we get a derived query less complex and comprehensible by the system.

Evaluation

To evaluate the system, we have to test it by some end users. This system was checked out by many naïve, intermediate and advanced users in different specialties (30 persons) to interact with it. Every one of them have different levels in using search engines following the quota mentioned earlier in this section.

The empirical results demonstrate that this system helps researchers to search very easily. Concerning help, the results show that 17% of experts, 80% of advanced and 100% of beginners need assistance in at least 7 tasks of 10 and 85% of total use the recommendation.

The statistics of 30 researchers exhibit an average of approximately one of 10 experts, 8 of 10 advanced users, and all of 10 beginners needed help, then 21 of total used our recommendations. The study provides evidence that a usage of simple search engine can be in fact a scenario of sufficient complexity to be of value, especially for novices. However, every user needs assistance with complex search or with rarely searched information.

The system is not only successful in resolving references to the previously provided information, but also acts actively and brings relevant questions depending on the dialog context within an optimal time.

As the relevance of a recommendation is subjective to each one, it is not possible to automate the assessment process [17].

The outcomes are quite encouraging. Some measures as precision, recall and F-Measure are illustrated in **Figure 5**. The best accomplishment was the ability of the system to estimate 8 out of 10 recommendations, which allowed an achievement of 82% precision (user N.A). Concerning the poorest performance, it was 4 perfect propositions out of 10 (users I.E and A.H with precision 67%), documented for advanced users. Additionally, recalls are typically over 90%. For a better analysis of the performance of the proposed system, the presented results were compared with the results obtained previously. The results are illustrated

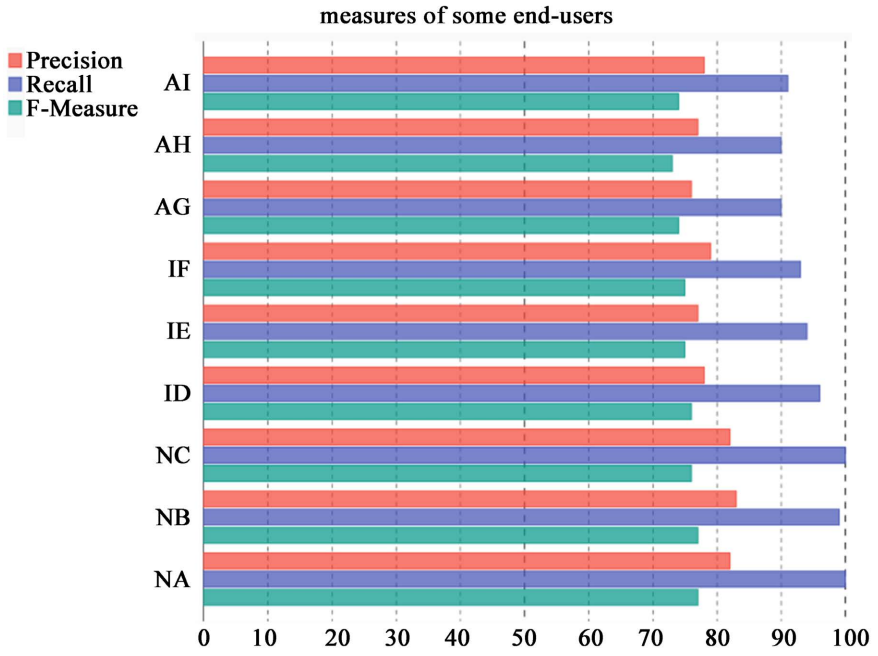

Figure 5. Precision and recall for some end-users.

Table 1. Results of Precision and recall for some users.

Users	Precision	Recall	F-measure
N.A	82%	100%	77%
N.B	83%	99%	77%
N.C	82%	100%	76%
I.D	68%	96%	76%
I.E	67%	94%	75%
I.F	79%	93%	75%
A.H	76%	90%	74%
A.I	67%	90%	73%
	78%	91%	74%

through the Table 1. Where A.G, A.H A.I represent advanced users, I.D, I.E, I.F are intermediate users and finally N.A, N.B and N.C was Naïve users.

Consequently, an assessment of the whole performance of the system was required from the participants; above 80% of them considered the system as assisting, and 70% came to the recognition that it is effective.

5. Conclusions and Future Work

In this paper, we have presented a coupled of item based collaborative filtering and content-based filtering recommender system operated by natural language interface and enhanced by machine learning algorithms such as Apriori and spanning trees. Recommender systems when coupled with natural language un-

derstanding become very promising techniques to retrieve and recommend relevant items. Our recommender system deals with an interesting issue of bibliographic research systems based on recommender system and natural language understanding. Moreover, the dialog system handles references to previous results and phrases, allowing the construction of different sentences. Our experimental results were very promising. Recommender systems based on content are generally subject to problems and persons such as overspecialization, because they try to find content using their syntactic similarity for an item [17].

For our future work, we plan to use the internal speech recognition tool of the Operating System to improve our recommender system. With these techniques, we will get a complete recommender system based on natural language, both written and spoken using machine learning optimizing algorithms.

References

[1] Lal, N., Qamar, S. and Shiwani, S. (2016) Information Retrieval System and Challenges with Dataspace. *International Journal of Computer Applications*, **147**, 23-28. https://doi.org/10.5120/ijca2016911128

[2] Osadchiy, T., Poliakov, I., Olivier, P., Rowland, M. and Foster, E. (2018) Recommender System Based on Pairwise Association Rules. *Expert Systems with Applications*, **115**, 535-542. https://doi.org/10.1016/j.eswa.2018.07.077

[3] Chakraoui, M. and El Kalay, A. (2016) Efficiency of Indexing Database Systems and Optimising Its Implementation in NAND Flash Memory. *International Journal of Systems, Control and Communications*, **7**, 221-239. https://doi.org/10.1504/IJSCC.2016.077406

[4] Chakraoui, M. and El Kalay, A. (2016) Optimization of Local Parallel Index (LPI) in Parallel/Distributed Database Systems. *International Journal*, **11**, 2755-2762. https://doi.org/10.21660/2016.27.1322

[5] Chakraoui, M., El Kalay, A. and Mouhni, N. (2016) Tuning Different Types of Complex Queries Using the Appropriate Indexes in Parallel/Distributed Database Systems. *International Journal*, **11**, 2267-2274. https://doi.org/10.21660/2016.24.1392

[6] Lika, B., Kolomvatsos, K. and Hadjiefthymiades, S. (2014) Facing the Cold Start Problem in Recommender Systems. *Expert Systems with Applications*, **41**, 2065-2073. https://doi.org/10.1016/j.eswa.2013.09.005

[7] Brusilovsky, P., Kobsa, A. and Nejdl, W. (2007) Data Mining for Web Personalization. Springer-Verlag, Berlin, 90-135.

[8] Tso-Sutter, K.H.L., Marinho, L.B. and Schmidt-Thieme, L. (2008) Tag-aware Recommender Systems by Fusion of Collaborative Filtering Algorithms. *ACM SAC'08*, Ceara, 16-20 March 2008, 1995-1999. https://doi.org/10.1145/1363686.1364171

[9] Agrawal, R. and Sricant, R. (1994) Fast Algorithms for Mining Association Rules. *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases*, Santiago de Chile, 12-15 September 1994, 487-490.

[10] Deshpande, D.S. (2014) A Novel Approach for Association Rule Mining using Pattern Generation. *International Journal of Information Technology and Computer Science*, **11**, 59-65. https://doi.org/10.5815/ijitcs.2014.11.09

[11] Amin, H., Devi, A. and Ul Amin, N. (2019) Predictive Analysis of Heart Disease Using K-Means and Apriori Algorithms. *JASC: Journal of Applied Science and*

Computations, **6**, 2183-2189.

[12] Pop, P.C. (2020) The Generalized Minimum Spanning Tree Problem: An Overview of Formulations, Solution Procedures and Latest Advances. *European Journal of Operational Research*, **283**, 1-15. https://doi.org/10.1016/j.ejor.2019.05.017

[13] Kruskal, J.B. (1956) On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, **7**, 48-50. https://doi.org/10.1090/S0002-9939-1956-0078686-7

[14] Biswas, P., Goel, M., Negi, H. and Datta, M. (2016) An Efficient Greedy Minimum Spanning Tree Algorithm Based on Vertex Associative Cycle Detection Method. *Procedia Computer Science*, **92**, 513-519. https://doi.org/10.1016/j.procs.2016.07.376

[15] Zhu, Y.J., Yan, E. and Song, I.-Y. (2017) A Natural Language Interface to a Graph-Based Bibliographic Information Retrieval System. *Data & Knowledge Engineering*, **111**, 73.

[16] Li, F. and Jagadish, H.V. (2014) Constructing an Interactive Natural Language Interface for Relational Databases. *Proceedings of the VLDB Endowment* **8**, 73-84. https://doi.org/10.14778/2735461.2735468

[17] Carrer-Neto, W., Hernández-Alcaraz, M.L., Valencia-García, R. and García-Sánchez, F. (2012) Social Knowledge-Based Recommender System. Application to the Movies Domain. *Expert Systems with Applications*, **39**, 10990-11000. https://doi.org/10.1016/j.eswa.2012.03.025

[18] Tuzhilin, A. and Adomavicius, G. (2015) Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, **17**, 734-749. https://doi.org/10.1109/TKDE.2005.99

[19] Sujatha, B. and Raju, D.S.V. (2016) Ontology Based Natural Language Interface for Relational Databases. *Procedia Computer Science*, **92**, 487-492. https://doi.org/10.1016/j.procs.2016.07.372

[20] Wachtel, A., Weigelt, S. and Tichy, W.F. (2015) Initial Implementation of Natural Language Turn-Based Dialog System. *Procedia Computer Science*, **84**, 49-56. https://doi.org/10.1016/j.procs.2016.04.065