DRHTG: A Knowledge-Centric Approach for Document Retrieval Based on Heterogeneous Entity Tree Generation and RDF Mapping

M. Arulmozhi Varman[1] and Gerard Deepak[2(✉)]

[1] Department of Electronics and Electrical Engineering, SASTRA Deemed University, Thanjavur, Tamil Nadu, India
[2] Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, India
gerard.deepak.christuni@gmail.com

Abstract. There is an enormous amount of data stored on the World Wide Web throughout the years of internet practice. There is a necessity for sophisticated techniques of acquiring and categorizing information resources available on the internet. In the domains of information retrieval and natural language processing, document retrieval has been an important task. Despite the fact that document retrieval algorithms have evolved significantly over the years, there is always space for improvement. Tokenization, lemmatization, SWE, and NER are among the preparation methods applied to the dataset. The preprocessed dataset is then transferred to XML format, where the RDF mapping is performed using the Jellyfish method. The knowledge graph is built using blogs, social network APIs, crowdsource community ontology, wiki data and Freebase, and LODC. The formulated knowledge graph is converted to RDF format from OWL format. The Jellyfish method is used on the updated knowledge graph in OWL format and the RCVI dataset in OWL format to conduct RDF mapping in terms of (subject predicate object predicate). To rank the document in increasing order of semantic similarity, KL divergence is applied to the RDF mapping obtained in the previous phase. Based on the increasing order of semantic similarity the most relevant document is recommended. The proposed framework accomplished an accuracy of 92.78%.

Keywords: Jellyfish · KL divergence · Optimization · RDF mapping · Text mining

1 Introduction

The comparison of a specified user's query against a collection of free-text data is termed document retrieval. The document retrieval application consists of a database of documents categorized according to the user queries and a user interface to access all these documents. The document retrieval has two main tasks: comparing the user query and categorizing the documents rank-wise from most relevant documents to less similar

A. Abraham et al. (Eds.): IBICA 2022, LNNS 649, pp. 414–424, 2023.
https://doi.org/10.1007/978-3-031-27499-2_39

documents. The relevant document is suggested to the user based on the order of semantic similarity [1]. This paper proposes a novel approach incorporating a heterogeneous entity tree generation scheme with RDF mapping to retrieve documents from the database [2]. RDF mapping is created using Jellyfish algorithm and the documents are re ranked in increasing order of semantic similarity by KL divergence [3, 4]. The re ranked documents is provided to the user.

Motivation: Since the beginning of recorded language, humans have been using various methods to quickly index and retrieve information. The evolution of technology has allowed us to store and search vast amounts of information. This paper explores some of the more advanced methods of document retrieval. Semantic-based document retrieval is a method that uses semantic information to aid in document retrieval.

Contribution: Document retrieval has been a significant task in the domain of information retrieval and natural language processing. Although document retrieval algorithms have undergone a significant transformation over the decades, there is still room for developing a better algorithm. The RCVI dataset undergoes various preprocessing procedures such as tokenization, lemmatization, SWE, NER. The preprocessed dataset is then converted into XML format, on which the Jellyfish algorithm is employed to perform RDF mapping. Blogs, social network API, crowd source community ontology, wiki data and Freebase, and LODC are used to formulate the knowledge graph. The knowledge graph formulated is converted into OWL format and further converted into RDF format. Jellyfish algorithm is employed on the modified knowledge graph in OWL format and the OWL format of the RCVI dataset to perform RDF mapping in terms of (subject predicate object predicate). KL divergence is used on the RDF mapping obtain in the previous step to rank the document in the increasing order of semantic similarity. Based on the document rank, the most relevant document is chosen and recommend to the user.

Organization: The paper is divided into five other sections. Section one comprises of introduction. Section two depicts the related works. The proposed architecture is discussed in section three. The implementation and performance evaluation is given in section four. Section five consists of results. Finally, the paper is concluded in section six.

2 Related Work

Ramya et al. [1] proposed DRDLC framework is proposed to provide/retrieve more relevant digital data over the web, which is dispersed globally and unorganized. This framework works based on the similar of the keywords from search results. It extracts high relevant documents to the user's query. Deka et al. [2] determined that KNN is one of the classification methods of information retrieval, which considers all samples and results in high computational complexity. Hence, to overcome the difficulty, the K-means clustering algorithm will group all the samples. The cluster centers are considered new samples in which KNN and Decision Tree are applied further to find out the documents'

category and Sub-category of the documents, respectively. Hershey et al. [3] proposed a method for improving the KL divergence between two generalized methods of movements (GMMs) is proposed. We then introduce two new methods for the improvement of the correctness of the results. Chou et al. [4] depicted the behavior of jellyfish in the water that inspires the development of a unique metaheuristic algorithm named as artificial Jellyfish search(JS) optimizer. In solving mathematical benchmark functions, the JS optimizer outperformed other methods. Li et al. [5] proposed semantic distance based Concept similarity is proposed to calculate concepts similarity effectively The distance is considered thoroughly and the semantic contact ratio and the depth differences between concepts. Shannon et al. [6] introduced a theory to incorporate a number of additional elements, including the influence of noise in the channel and the savings achievable owing to the statistical structure of the original message and the nature of the information's eventual destination. Cilibrasi et al. [7] method is based on the idea of similarity complexity between terms and phrases based on the information distance and Kolmogorov complexity. First, to fix thoughts, we use www and any search engines and databases. Then, it is applied to construct a method to automatically extract similarity (Eg. Google page counts). This paper introduces an automatic method to extract similarities from the text. It can also perform an English-Spanish translation. Finally, using WordNet categories which is crafted by the experts is resulted from a massive randomized trial in binary classification using support vector machines. Kuzi et al. [8] proposed Hybrid strategies combine the semantic retrieval model with the linguistic approach and models in semantic retrieval. This study shows how effective our method is and provides insight into the various aspects of the semantic approach. Karami et al. [9] study uses a systematic strategy to mine many Twitter-based studies to describe the relevant literature. This study gathered relevant articles from three databases and used text mining and trend analysis to discover semantic trends and investigate the annual evolution of research themes over a decade. Ensan et al. [9] proposed a semantic-aware language model that addresses the challenge of document relatedness in text mining. The model consists of a graph representation of concepts retrieved from text and nodes representing semantic relatedness. Antons et al. [10] proposed a set of prioritized development goals that will help improve the quality of text mining. Lui et al. [11] analyzed the XML tree structure and separated the XML components into three sub models based on the structure. They also abstract the aggregates into abstract structure models. In [12–17] several Ontology based approaches in support of the proposed approach has been discussed.

3 Proposed Work

Figure1 depicts the architecture of the proposed approach for text mining from documents using an RDF-driven approach. The approach consists of three phases. The first phase involves the preprocessing of the dataset. The RCVI dataset used for experimentation is subject to preprocessing, which involves tokenization, lemmatization, stop word removal, name entity recognition. The inflectional form of the terms within the documents and contents is converted into a custom structured XML file creation. The XML is created mainly for converting to RDF or the resource description framework as direct formulation of RDF is impossible, so an intermediate XML structuring is required. XML

to RDF conversion is achieved using the ontocolab. At the end of phase one, the dataset is preprocessed. A structured RDF file from an intermediate XML structured file is created for all the documents available in the dataset.

The second phase is the generation of the knowledge graph. The knowledge graph is generated by using the categories and the clue tag categories and the domains in the dataset, and several terms from existing user-contributed blocks; the Twitter API is used to extract the set of terms matching in the categories in the RCVI dataset. The categorized RCVI dataset and the terms set generated from blogs and used in Twitter API are subjected to cloud source community ontology (OWL) files available on the current structure of the World Wide Web to yield much more concentrated knowledge generation. All the terms generated from the blogs and cloud source community ontology (OWL) are subjected to querying of the wiki data and the free base through agents to yield much more dense knowledge. Further, all the terms which are extracted from the categories of the RCVI dataset, as well as the domains in the RCVI dataset and terms generated from the blogs, Twitter API, cloud source community ontology (OWL), wiki data, and free base, are all subjected to querying in LODC cloud to generate fragments of sub graphs and all the sub graphs, as well as the tag terms and the trees generated from each of the processes, are collectively formulated into knowledge graph through establishing the relation between the nodes either by semantic similarity analysis or topic mapping.

Phase three involves converting the knowledge graph before it is mapped with the existing RDF file, which is generated from the dataset that is not in the appropriate format for RDF conversion. Once the knowledge graph is also subjected to conversion to RDF, the RDF-RDF mapping takes place. Mapping takes between the subject-subject mapping between the existing RDF generated from phases 1 and 2, then object-object mapping and the predicate analysis. Maybe the subject-subject mapping, object-object mapping, subject-object mapping, and object-subject mapping are considered by computing the KL divergence and Shannon's entropy. Mainly predicate analysis is just a subsidiary process. It can be ignored, neglected, and not considered as subject-subject mapping, object-object mapping, subject-object mapping, and object-subject mapping are much more efficient.

Once the documents are mapped and re-ranked by increasing order of the semantic similarity. Subject-subject mapping, object-object mapping, subject-object mapping, and object-subject mapping are done by computing the KL divergence, Shannon's entropy, and semantic similarity computation. The KL divergence value must be less than 0.25, and for entropy, it should be greater than 0.5, and the semantic similarity threshold must be greater than 0.75. The intersection between Lin similarity and normalized Google distance (NGD) and concept similarity will be the semantic similarity. The reason for generating a lot of content from the blog, Twitter API, cloud source ontology, wiki data, free base, and LODC is to increase the number of entities exponentially so that the knowledge graph and extensive knowledge graph can be formulated. RDF mapping is considered rather than an OWL-based ontology mapping is mainly because the specificity between the entities terms in RDF-based mapping is much higher when compared to the traditional stand-alone OWL-based ontology mapping. As a result, there is

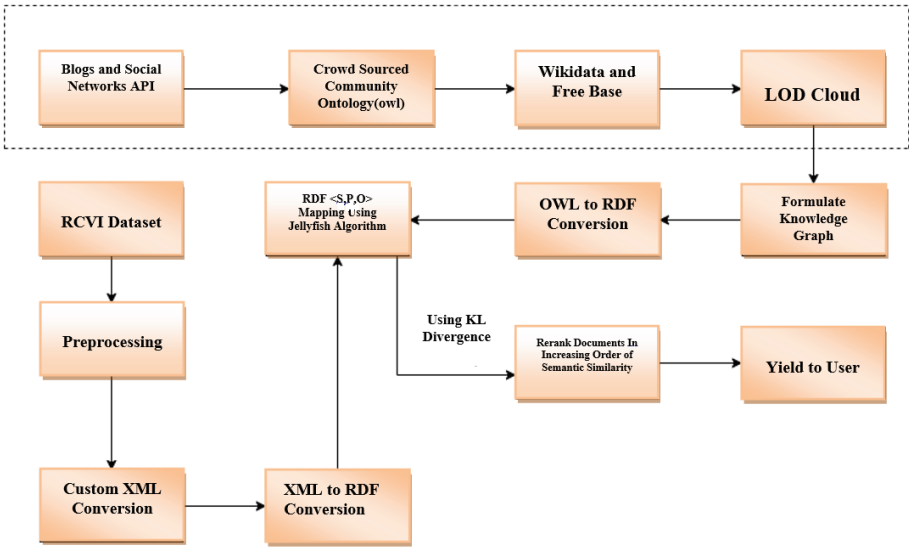

Fig. 1. Architecture flowchart of the proposed DRHTG model.

a much better scope when compared to the traditional ontology matching or semantic similarity-based analysis.

4 Implementation and Performance Evaluation

4.1 Dataset Preparation

The RCV1 and RCV2 datasets comprise a register containing five sub-register in five languages: English, French, German, Italian, and Spanish. Each sub-register in these five languages contains five files, each having directories of the files translated or written in the respective language. In the English sub-register, there are five files translated from each to English respectively and similarly for four other languages. All directories have the same number of documents in the same order, and the number of lines will also be the same.

Algorithm 1. Algorithm for the proposed DRHTG Framework

Input: RCVI dataset.
Output: Most relevant documents to the user's query.

Start
Step 1: The dataset is preprocessed: tokenization, lemmatization, stop words removal, word entity recognition.
Step 2: The preprocessed dataset is converted into an XML format.
Step 3: The dataset in XML is converted to RDF format.
Step 4: RDF graph is converted into the corresponding RDF mapping represented in the form of subject object query predicate.
Step 5: A knowledge graph is formulated from various blogs and social networks API, crowd source community ontology in OWL format, wiki data and free base and LODC.
Step 6: The formulated knowledge graph is in the OWL format and it is converted into RDF format.
Step 7: The formulated knowledge graph inn RDF format undergoes RDF mapping.
Step 8: Upon RDF mapping the jelly fish algorithm is employed along with KL divergence to rank the documents in the increasing order of semantic similarity.

```
Create a starting occupants Y = {Y_i,Y_i,…., Y_m}
 Evaluate the volume of food at respective position Y_i by f(Y_i)
 Observe the great answer of the starting occupants Y'_great
 For x = 1 to Iter_max do
        For j = 1 to n_occ do
                Formulate the time control value D(t)
                If D(t) ≥ D_o
                        Jellyfish go along with ocean current
                Else
                        Jellyfish goes into a swarm
                        If rand [0,1] > [1-D(t)]
                                Jellyfish displays passive shifting
                        Else
                                Jellyfish displays active shifting
                        End if
                End if
                Examine border state at a different position Y_i
                Evaluate the volume of food at respective position f(Y_i)
                Improve the best answer
          End for
 End for
```

Step 9: Based on the rank the most relevant document is fetched to the user.
End

The Normalized Google Distance (NGD) between two search terms u and v is shown in Eq. (1). Where N is the overall amount of Google-searched internet sites multiplied by the average number of singleton target keywords found on those pages; $f(u)$ and $f(v)$ are

the amount of similar searches for terms u and v, respectively; and F (u, v) is the number of internet pages that include both a and b. The conditional entropy of two variables can also be defined as X and Y taking values x_i and y_j respectively is depicted in Eq. (2). Where $p(x_i, y_j)$ is the probability that $X = x_i$ and $Y = y_j$. This value should be interpreted as the amount of randomness in the random variable X given the random variable Y. The Kullback-Leibler Divergence score, often known as the KL divergence score, measures how one probability distribution differs from another which is shown in Eq. (3). The KL divergence is the negative sum of each event's probability in P multiplied by the log of the event's probability in Q over the probability of the event in P. Lin Similarity is engaged to estimate the degree of the semantic relationship between units of concepts, language, and instances. The Lin Similarity is calculated as the ratio of the similarity between the terms upon the difference between them, i.e., the commonality and difference ratio. It can be formulated, as shown in Eq. (4). The similarity of concepts and terms in terms of text and query is a reflection of the degree to which semantic matching exists between them. Similarity between concept Ci and Cj denoted by $Sim(C\text{i}, C\text{j})$ in the Eq. (5).

$$NGD(u, v) = \frac{\max\{logf(u), \log f(v)\} - logf(u, v)}{logN - \min\{logf(u), logf(v)\}} \tag{1}$$

$$H(X|Y) = -\sum_{i,j}^{n} P(x_i, y_j) \log \frac{p(x_i, y_j)}{p(y_j)} \tag{2}$$

$$D_{KL}(P\|Q) = \sum_{x \epsilon \chi} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{3}$$

$$sim_{Lin}(X, Y) = \frac{\log P(comman(X, Y))}{\log P(description(X, Y))} \tag{4}$$

$$sim(C_i, C_j) = \alpha * sim(C_i, C_j)_{dist} + \beta * sim(C_i, C_j)_{const} + \gamma * sim(C_i, C_j)_{cdepth} \tag{5}$$

5 Results

Data is collected from Freebase and wiki data, Crowd sources community ontology, Blogs and social networks API and LODC cloud which it then formulated into knowledge graph. The formulated graph is then converted to RDF format from OWL format. Jellyfish algorithm is used for RDF conversion. RCVI dataset is preprocessed into custom XML format then converted to RDF format. Then using KL divergence both the converted RDF mapping is re ranked in increasing order of semantic similarity (Fig. 2).

Table 1 compares the performance of the proposed model with the base model approach and other approaches. Figure 3 compares the precision percentage of the baseline models with DRHTG.

It is evident from Table1 that the proposed framework DRHTG operates better than LSI+Cosine +Jaccard, MLAP, STM+Cosine, DRDLC, and LSI+Fuzzy C-means Clustering. Precision, F-measure, FDR, nDGC, and accuracy have all improved in the proposed DRHTG. The F-measure value of DRHTG is greater than LSI+Cosine +Jaccard, MLAP, STM+Cosine, DRDLC, and LSI+Fuzzy C-means Clustering by 14.22% 12.37%,

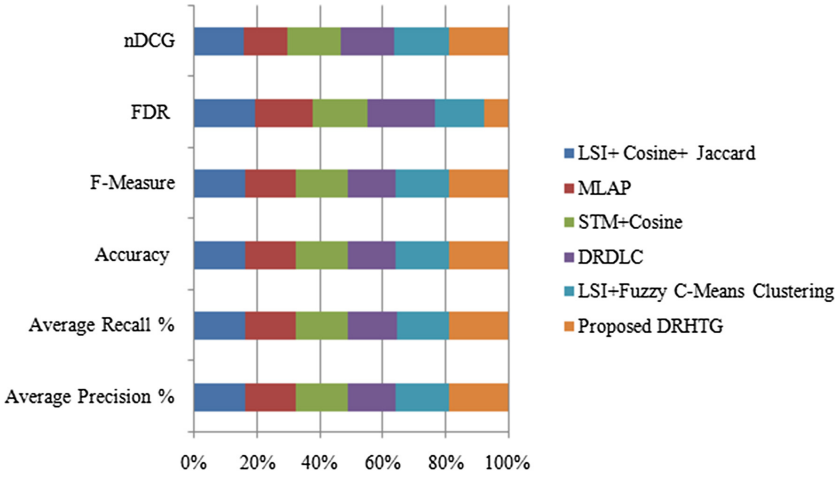

Fig. 2. Precision percentage vs No. of recommendations

Table 1. Comparison of Performance of the proposed DRHTG with other approaches

Search Technique	Average Precision %	Average Recall%	Accuracy	F-Measure	FDR	nDCG
LSI+Cosine+Jaccard	77.32	79.81	78.56	78.54	0.23	0.79
MLAP [2]	78.71	82.15	80.43	80.39	0.22	0.69
STM+Cosine	79.11	83.15	81.13	81.07	0.21	0.86
DRDLC [1]	74.93	78.18	76.55	76.52	0.26	0.86
LSI+Fuzzy C-Means Clustering	81.18	84.17	82.675	82.64	0.19	0.87
Proposed DRHTG	91.78	93.78	92.78	92.76	0.09	0.95

11.69%, 16.24%, and 10.12%, respectively. The proposed model's precision is 91.78%, while LSI+Cosine +Jaccard, MLAP, STM+Cosine, DRDLC, and LSI+Fuzzy C-means Clustering are 77.32%, 78.71%, 79.11%, 74.93%, and 81.18, respectively. The accuracy of DRHTG is greater than that of LSI+Cosine +Jaccard, MLAP, STM+Cosine, DRDLC, and LSI+Fuzzy C-means Clustering by 14.22%, 12.35%, 11.65%, 16.23%, and 10.12%. The recall of the proposed approach is better than LSI+Cosine +Jaccard, MLAP, STM+Cosine, DRDLC, and LSI+Fuzzy C-means Clustering in percentage by13.97, 11.63, 10.63, 15.6, and 9.61. The nDGC of the proposed approach is lower than LSI+Cosine +Jaccard, MLAP, STM+Cosine, DRDLC, and LSI+Fuzzy C-means Clustering by 0.16, 0.26, 0.09, 0.09, and 0.08, respectively. The FDR values of LSI+Cosine + Jaccard, MLAP, STM+Cosine, DRDLC, and LSI+Fuzzy C-means Clustering are greater than DRHTG by 0.14, 0.13, 0.11, 0.17, and 0.1.

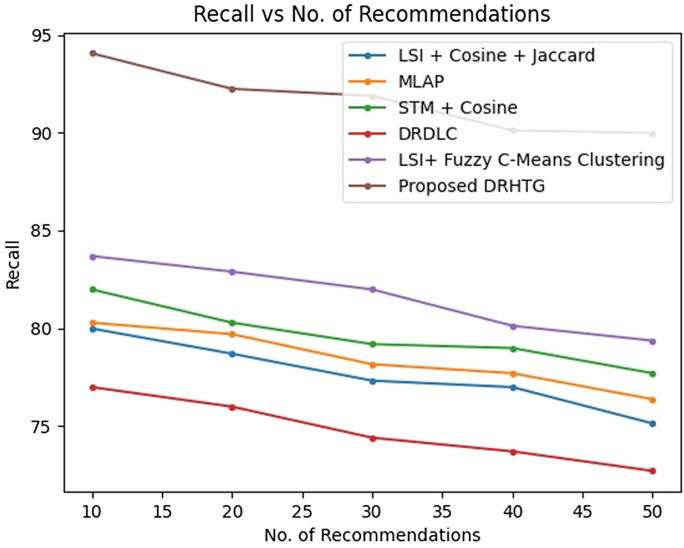

Fig. 3. Pictorial depiction of the proposed DRHTG and other baseline models

6 Conclusion

The results' efficacy validates the proposed framework's ability to be used for document retrieval. The RCVI dataset obtained is preprocessed to obtain custom RDF mapping. Blogs and social networks API, Crowdsource community ontology, wiki data, free base, and LODC cloud are incorporated into the RDF conversion to improve classification accuracy. Upon classification, Jellyfish Optimization is employed on the classified results to re-rank the documents in increasing order. The proposed DRHTG framework yields an average accuracy of 92.78%, with a very low FDR of 0.09. Also, the Jellyfish Optimization algorithm has been employed on the classified results to yield more accurate results. The results validate that DRHTG is the best-in-class approach to re-rank documents by semantic similarity and yield it to the user.

References

1. Ramya, R.S., Sejal, D., Venugopal, K.R., Iyengar, S.S., Patnaik, L.M.: DRDLC: discovering relevant documents using latent dirichlet allocation and cosine similarity. In: Proceedings of the 2018 VII International Conference on Network, Communication and Computing, pp. 87–91, 14 Dec 2018

2. Deka, H., Sarma, P.: Machine learning approach for text and document mining. Int. J. Comput. Sci. Eng. (IJCSE). **6**(5) (2017)
3. Hershey, J.R., Olsen, P.A.: Approximating the Kullback Leibler divergence between Gaussian mixture models. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP2007. Vol. 4, pp. IV-317. IEEE 15 Apr 2007
4. Chou, J.S., Truong, D.N.: A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. Appl. Math. Comput. **15**(389), 125535 (2021)
5. Li, W., Xia, Q.: A method of concept similarity computation based on semantic distance. Procedia Eng. **1**(15), 3854–3859 (2011)
6. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948)
7. Cilibrasi, R.L., Vitanyi, P.M.: The google similarity distance. IEEE Trans. Knowl. Data Eng. **19**(3), 370–383 (2007)
8. Kuzi, S., Zhang, M., Li, C., Bendersky, M., Najork, M.: Leveraging semantic and lexical matching to improve the recall of document retrieval systems: a hybrid approach. arXiv preprint arXiv:2010.01195. 2 Oct 2020
9. Karami, A., Lundy, M., Webb, F., Dwivedi, Y.K.: Twitter and research: a systematic literature review through text mining. IEEE Access. **26**(8), 67698–67717 (2020)
10. Antons, D., Grünwald, E., Cichy, P., Salge, T.O.: The application of text mining methods in innovation research: current state, evolution patterns, and development priorities. R&D Manage. **50**(3), 329–351 (2020)
11. Liu, Y., Hong, Z.: Mapping XML to RDF: an algorithm based on element classification and aggregation. In: Journal of Physics: Conference Series. Vol. 1848, no. 1, p. 012012. 1 Apr 2021 IOP Publishing
12. Arulmozhivarman, M., Deepak, G.: OWLW: ontology focused user centric architecture for web service recommendation based on LSTM and whale optimization. In: Musleh Al-Sartawi, A.M.A., Razzaque, A., Kamal, M.M. (eds.) EAMMIS 2021. LNNS, vol. 239, pp. 334–344. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77246-8_32
13. Surya, D., Deepak, G., Santhanavijayan, A.: KSTAR: a knowledge based approach for socially relevant term aggregation for web page recommendation. In: Motahhir, S., Bossoufi, B. (eds.) ICDTA 2021. LNNS, vol. 211, pp. 555–564. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73882-2_50
14. Surya, D., Deepak, G., Santhanavijayan, A.: QFRDBF: query facet recommendation using knowledge centric DBSCAN and firefly optimization. In: Motahhir, S., Bossoufi, B. (eds.) ICDTA 2021. LNNS, vol. 211, pp. 801–811. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73882-2_73
15. Surya, D., Deepak, G., Santhanavijayan, A.: Ontology-based knowledge description model for climate change. In: Abraham, A., Piuri, V., Gandhi, N., Siarry, P., Kaklauskas, A., Madureira, A. (eds.) ISDA 2020. AISC, vol. 1351, pp. 1124–1133. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-71187-0_104

16. Deepak, G., Santhanavijayan, A.: QGMS: a query growth model for personalization and diversification of semantic search based on differential ontology semantics using artificial intelligence. Comput. Intell. 1–30 (2022)
17. Deepak, G., Santhanavijayan, A.: OntoDynS: expediting personalization and diversification in semantic search by facilitating cognitive human interaction through ontology bagging and dynamic ontology alignment. J. Ambient Intell. Humanized Comput. 1–25 (2022)