RISL:RDF Driven Indexing Scheme Using Intelligent Semantics and Lion Optimization

Gerard Deepak[1], Arulmozhi Varman M[2], Palvannan S[2], Deepak Surya S[2], Dev Agrawal[2], Ashvanth R[2]

[1,2]Department of Computer Science and Engineering
[3]Department of Information
[1]Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education,Manipal, India
[2]National Institute of Technology, India
[3]University of Toronto, Canada
[1]gerard.deepak.cse.nitt@gmail.com

Abstract

Indexing documents and Web pages is a vital task in the modern-day world as the documents and Web pages are increasing on the World Wide Web. In this paper, the RISL which is a framework for the metadata driven RDF centered model for indexing Web documents has been proposed. The model is based on Entity Enrichment using the Structural Topic Model. The Subject and Object pairs generated in the RDF are retained and aggregated with the upper ontologies which not only provision lateral term pair semantics but also serves as an active model to increase the knowledge density. Furthermore. The encompassment of Googles Knowledge Graph API for entity enrichment based on knowledge graph synthesis, andthe classification of dataset using the ensemble.Bagging model which inturn integrates Support Vector Machine and Random Forest classifiers facilitate in anchoring the right knowledge for generation of indexes. The semantic similarity computation using the Adaptive Pointwise Mutual Information (APMI) measure and the deviance computing criteria using the Heips Evenness Index and the Pearson Correlation Coefficient at varied instances with different levels of heterogeneity ensures strong relevance computation schemes. The RISL framework encompasses the Lion Optimization scheme for refining the feasible solution set to derive optimal solution set for indexing Web documents. The proposed RISL framework yields an overall precision of 95.12 with the lowest FDR of 0.05 which is the best in class model for generating indexing of Web documents.

Keywords- RDF, Indexing, Intelligent Semantics, Lion Optimization, Semantic Similarity

1 INTRODUCTION

A piece of information's relevance is considered as part of the information retrieval process based on predetermined retrieval criteria. The size of the data on the World Wide Web worldwide is enormous. It represents a significant challenge for computer software developers trying to understand how best to search the Web. With the rapid climb of the Web, communication and information retrieval have become very easy using the World Wide Web. The main goal is to choose the best collection of information based on user requirements. In order to determine the frequency of words in Web documents, the focused crawlers currently in use employ various methods. The document is regarded as relevant if higher frequency words coincide with the subject keyword. This approach has a drawback because it loads the keyword without considering its context on the page. The topic of context has received a lot of attention in data recovery literature for examining pertinent data. For many Web applications, it is crucial to investigate the online page contents for automatic indexing. An index is stored in order to improve speed and performance and locate pertinent documents for an inquiry query. The program would have to scan every document in the corpus without an index, which could take a lot of time and processing power. The time saved during information retrieval is exchanged for the extra memory needed to store the index, which is also necessary due to the significant increase in the time it takes for an update to need a place. Indexing is a strategy where indexing servers mark documents to understand the context, meaning, or fit of the Web document and facilitate the access of documents by search engine indexing. Indexing is fundamental, especially for organizing data in the semantic Web and semantic 3.0. Indexing is a characteristic of semantic Web 2.0. A semantic Web or semantic 3.0 has the best fit open linked data, which is harnessed by knowledge indexing, becomes quite important in Web 3.0 like in Web 2.0. In Web 3.0, it is to harness the document with the existing lateral knowledge in terms of open linked data of the structure of the Web 3.0.

Motivation: The speedy hike of internet users and many people's dependency on the World Wide Web has made it an ultimate resource for obtaining information and knowledge. The Internet is the fastest means of communication media with no competitor behind. The Internet works like a gigantic library because of its interconnectivity. The rapid rise of internet users and the increasing dependence of people worldwide on the Web have made it a last resort for information and knowledge. The Internet is the fastest way to communicate with no one else competing in the background. The Internet acts as a great library because of its connection.

Contribution: The proposed framework for indexing Webpages has been proposed, which integrates the TF-IDF model and the structural topic modelling along with RDF generation. The RDF subject and object are retained and aggregated with upper ontologies for further enrichment. The incorporation of Google knowledge graph search API ensures strong entity-rich auxiliary knowledge into the model, the bagging feature control machine learning bagging classifier with support vector classifier and random forest as the independent constituent models for the ensemble model, and the proposed framework yields comprehensive results. The proposed semantic similarity is computed using APMI with heap evenness index and Pearson's correlation coefficient with differential thresholds and step deviation measures. This ensures that the proposed framework has robust relevance computations mechanisms in the model. The performance is evaluated using Precision, Recall, Accuracy, F-measure and FDR to benchmark the baseline models.

Organization: The structure of the article is as follows. Section 2 discusses the relevant works. In section 3, the proposed system architecture is described. Section 4 discusses implementation. Results and the performance analysis are displayed in section 5. Section 6 contains the paper's conclusion.

2 RELATED WORKS

Mukhopadhyay et al., [1] first retrieve the dominant and sub-dominating terms for Webpages that are taken from the domain-specific repository, then apply the appropriate secondary and primary attachment rules. Attia et al., [2] aim to develop a multi-criteria ranking and indexing model that considers the various factors that affect the quality of documents and pages. It aims to create a model that can achieve high performance and improve the relevance of both online and offline pages. Manjula et al., [3] proposed a new information retrieval method to provide the user with the most accurate information. This method would allow the user to access the data from the collection easily. Yazdani et al., [4] proposed this method by comparing it with some of the most popular and well-known optimization techniques. The development of this algorithm was motivated by the various characteristics of lions, such as their cooperation and lifestyle. Gao et al., [5] developed a support vector classifier (SVC) based on gradient descent suitable for binary classification. Two structures are proposed for the method: the dual and the primitive forms. The training of the SVC is performed on a dual-form basis. A comprehensive comparison of the various training methods is conducted. Wright et al., [6] discussed 16 topics in this study that revolved around various aspects of depression. The most common strategy was to think positively, focusing on positive aspects of life. Other strategies included participating in activities and hobbies, keeping a consistent schedule, and focusing on one day at a time. Zhan et al., [7] were able to identify the critical features of the COVID-19 pandemic through the use of random forest and then combined the strategy and the BLS to develop a forecasting model that considers the various factors that affect the pandemic and then compared the results with other models such as the linear regression model, the decision tree model, the adaptive boosting model, the RF, and the gradient boosting DT. The RF-Bagging model performed better in terms of forecasting performance when compared with other models. It also exhibited better statistical performance in terms of absolute and median absolute errors. Grootendorst et al., [8] showed that clustering techniques could be used to develop topic models. Topics can be helpful tools for finding hidden topics in documents. For instance, BERTopic can create clusters of document embeddings that pre-trained language models power. It can also generate topic representations using the class-based TF-IDF procedure. Liu et al., [9] aim to provide a comprehensive analysis of the various studies and discussions that have been conducted on this technology. The concept of named entity recognition has attracted widespread attention due to its potential to improve the efficiency of information extraction. Kumar et al., [10] proposed a security framework that combines the capabilities of artificial intelligence and deep learning. Using these techniques can give users full autonomy in their decisions. Second, it can also perform data load balancing and storage of IoT data through a distributed file system. Van Assche et al., [11] establish guidelines for producing knowledge graph platforms and a benchmark for the virtualization and materialization approaches. Through his research, he hoped to make the production of knowledge graph platforms more accessible to small and medium-sized enterprises. In [12-20] several models relevant to the literature of the proposed framework are depicted.

3 PROPOSED SYSTEM ARCHITECTURE

The proposed architecture for the semantically oriented RDF-driven indexing framework is shown in Figure 1. The proposed framework is dataset-driven, and the dataset is preprocessed using Named Entity Recognition (NER), stop word removal, lemmatization, and tokenization. The preprocessed dataset produces the categories, and then the categories are extracted from the datasets.As the first step, the data is cleaned and preprocessed. The text is extracted, and the punctuations are removed from them. The common filler words are also removed. Then it is subjected to tokenization and lemmatization. Tokenization is converting the text sentences into simple units such as words. Lemmatization is performed to extract the root words, so the lemma of these words can be fed to the model instead of the word itself. Upon preprocessing, the categories in the data are extracted. The TF-IDF is also applied to the preprocessed dataset to convert the text data into vectors so it can be input into a machine learning model. Structural Topic Modelling is performed on the vectors generated based on TF-IDF and the categories extracted in the previous step. Structural Topic Modelling incorporates metadata into the model, enabling the model to comprehend how different documents with different word choices could still hold the same topic. The TF-IDF is applied within the dataset to produce the most informative terms across the Web corpus and the most frequent terms within the Web document, in addition to extracting categories from the dataset, which is a URL or Web page corpus. The TF-IDF model calculates the terms that appear most frequently on the Web page and the ones that appear least frequently throughout the Web corpus. The structural topic modelling (STM) pipeline receives the entities and categories produced by the TF-IDF model for topic modelling. This topic modelling technique explicitly created with social science research is called the Structural Topic Model (STM). STM enables us to incorporate metadata into our model and reveal how various texts may discuss the same fundamental subject differently. Topic modelling ensures that the hidden and pertinent terms uncovered from the outside world corpus do not overlap with the knowledge that is part of the framework. However, topic modelling of STM alone is insufficient, and henceforth RDF is generated.

Semantic wikis and external Web corpora are used as the base environment for the OntoCollab, which generates RDF. A subject, predicate, and object make up the triadic format of the Resource Distribution Framework (RDF). However, a predicate is omitted due to the predicate's complexity and heterogeneous structure, effectively retaining only the RDF subject and object to provide a higher degree of co-occurrence and semantics. This results in the retention of the RDF subject and object, and the subject and object entities are treated as distinct entities. Upper ontologies are used to educate the generated RDF, and these upper ontologies are produced using two frameworks: Stardog and OntoCollabitself.The dataset's categories are used to generate upper ontologies, which are automatically generated semi-automatically—that is, and then double-checked by domain experts and knowledge engineers. Because they strictly retain the strict region only up to four levels and ignore it beyond that, they are known as upper ontologies. Only the first three levels are considered if the fourth level is not present. However, eradicating all instances only leaves the base classes and their subclasses with the higher ontologies.Upper ontologies were chosen because they improve the model's suitability and conceptual clarity. To produce entities, esteemed entities, and categories together, these aggregated upper ontologies are combined with those of the RDF and TF-IDF. These enriched categories are then submitted as features to the Bagging Classifier to classify the dataset. Bagging is an ensemble classifier that combines the SVC [Support vector classifier] and the random forest classifier, two robust independent classifiers.

Bootstrap aggregating, also known as bagging, is a Parallel ensemble method for reducing variance in prediction models by producing more data during the training phase. This was created through random sampling and set replacement. Some observations might be repeated in every new training data set if sampling with replacement is used. Every element in the Bagging situation has an equal chance of turning up in a new dataset. The final predictor, also called a bagging classifier, averages or votes (classifies) the forecasts provided by each estimator or classifier (regression), in this case, the SVC and Random Forest classifier. The Bagging classifier combines predictions from various estimators, reducing variance.

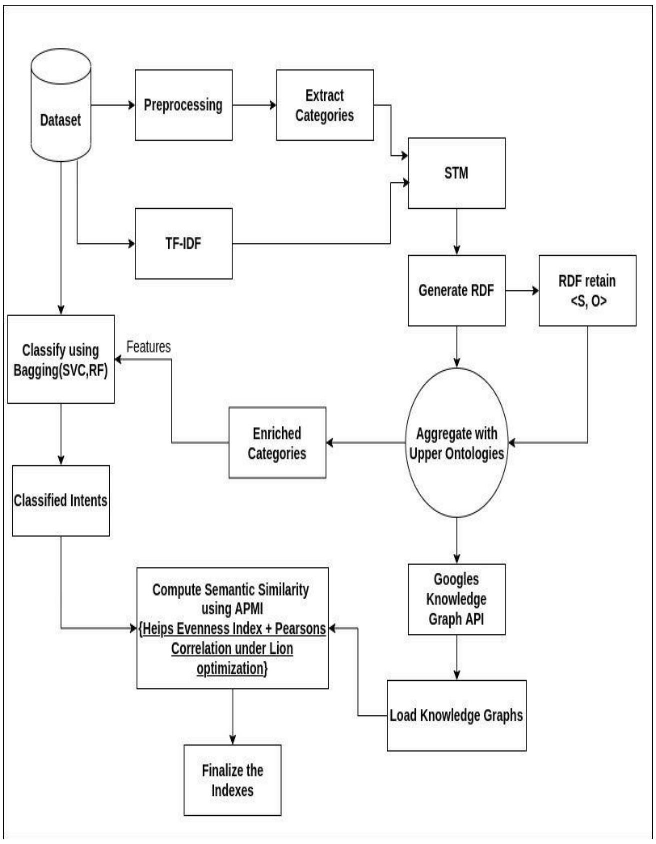
Dataset
Preprocessing
Extract Categories
STM
TF-IDF
Generate RDF
RDF retain <S, O>
Classify using Bagging(SVC,RF)
Features
Enriched Categories
Aggregate with Upper Ontologies
Classified Intents
Googles Knowledge Graph API
Compute Semantic Similarity using APMI {Heips Evenness Index + Pearsons Correlation under Lion optimization}
Load Knowledge Graphs
Finalize the Indexes

Fig.1. ProposedSystem Architecture

A simplistic linear SVM classifier draws a straight line between two classes. To put it another way, the data points on one side of the line will all be categorised into a single group, whereas the data points on the other side of the line will be categorised into a different group. This implies that the number of possible lines is unlimited. SVMs are distinct from other classification methods, and they choose the decision boundary that minimizes the distance from the nearest data points for all classes. The SVM-generated decision boundary is known as the maximum margin classifier or maximum margin hyperplane. Random Forest is a classifier that utilizes numerous decision trees on various subsets of the input data and averages the outcomes to improve the predicted Accuracy of the dataset. As long as they don't constantly all error in the same direction, the trees protect each other from their individual errors, which is why random forest was chosen.

A classified instance, the result of the bagging classifier, is then used in the model. The combined upper ontologies are then added to RDF, which already contains terms derived from STM categories and TF-IDF, by passing them through the Google knowledge graph API to load knowledge graphs, and knowledge subgraphs are pertinent to the combined upper ontologies. To load the knowledge graph and knowledge subgraph, all these terms are sent to the Google knowledge graph API. The semantic similarity of instances that have been classified is calculated using these knowledge graphs. The semantic similarity between the classes was then calculated.The top five instances are selected each time for each matching class, and the semantic similarity is computed using the APMI measures, Pearson correlation coefficient, and Heip's evenness index. The threshold is set at 0.75 for the APMI Measure and 0.35 for the Heip's evenness index and Pearson correlation coefficient. It's necessary to finalize the entities that are the most secure. However, because there are so many instances in the knowledge graphs, it only produces the initial solution. Additional optimization is necessary. The original solution is now referred to as the feasible solution. The optimization algorithm passes it through the Lion optimization algorithm, which transforms a much more optimal solution into an extinct feasible solution to finalize the indexes. The Step Deviance was computed at varied instances using the Heip's Evenness Index which is depicted in Equation (1) and Pearson's Correlation Co-efficient depicted in Equation (2) where H' is Shannon's index of diversity and S is species richness.

$$HE = \frac{\exp\ \hat{}\{H\prime\}}{S-1} \quad (1)$$

$$r = \frac{n\left(\sum\ xy\right)-\left(\sum\ x\right)\left(\sum\ y\right)}{\sqrt{\left[n\sum\ x^2-\left(\sum\ x\right)^2\right]\left[n\sum\ y^2-\left(\sum\ y\right)^2\right]}}$$

(2)

$$APMI(a;b) = \frac{pmt(a;b)}{p(m)+(n)} + k \qquad (3)$$

Where,r is Pearson Coefficient, n= number of the pairs of the stock, ∑xy implies sum of products of the paired stocks, ∑x indicates sum of the x scores, ∑y is the sum of the y scores∑x^2 is the sum of the squared x scores and ∑y^2 values the sum of the squared y scores.To calculate semantic similarity, the APMI measure is utilized. The Pointwise Mutual Information (PMI) measure has been modified to APMI. The APMI shown in Eq. (3) has an additivity coefficient k and performs significantly better than the other PMI variations.

4 IMPLEMENTATION

Python was used to implement our suggested framework on an Intel Core I5 computer with 16 GB of RAM in a Google collaborative environment at a high speed of 3.2 Giga Hertz. The experimentation was done in Python 3.10 using a combined integrative dataset that included Web search and spatial associative aspects of the Web [12] as well as the English language Web page dataset made available by Figshare [13]. These datasets were Web text datasets that were integrated, further annotated, combined, and customized into a new dataset, which was then used as the basis for additional experiments.

Algorithm 1: Proposed RISL Algorithm for RDF Driven Indexing:

```
Input: User queries, Upper Ontologies

Output: RDF driven indexes that are relevant t

Start

Step 1: The input queries are formalised to yield pre-processed query terms.

Step 2: The dataset is preprocessed using tokenization, lemmatization, word entity
recognition and stopword removal.

Step 3: Compute TF-IDF for the documents in the dataset

Step 4: Extract the class categories from the preprocessed data

Step 5:Employ structural topical modelling to extract RDF

Step 6: Aggregate the generated RDF with upper ontologies to enrich the categories

Step 7: Classify the features extracted from the enriched categories and the preprocessed data
using bagging with support vector machine and Random Forest algorithm.

        Random forest algorithm:

        Start

          Step 1: Initializer the parameters

          Input the variable X consisting of a set of features.

          Input variable Y which is the target variable.

          Store these two variables in S.

          F is the number of features

          Initialise H as NULL

          For  each i in 1 to n (number of trees)

          Step 2: Cal function Randomise (S,F)

          H <- H U {h(i})

          End of the loop

         End
```

```
    Support Vector Machine algorithm:
    Start
      condidateSv={closet pair from opposite classes}
      While there are violating points do
      Find a violator
      candidateSV = U candidateSV
      S
      Violator
      If any <0 due to addition of c to S then candidateSV = candidateSV \p repeat till all
      such points are pruned
          end if
    end while
    End
Step 8:Load the knowledge graph from the RDF using Google KG API
Step 9: Compute semantic similarity among the classified entities and the knowledge graph
entities and using APMI measure.
Step 10: Finalise the indices based on the semantic similarity score
End
```

Upper ontologies and user queries are inputs, and the structured queries provide preprocessed terms. Tokenization, lemmatization, word entity recognition, and stopword removal are part of the preprocessing. Additionally, calculating the TF-IDF for each document in the dataset is done, followed by using the preprocessed data to extract the class categories. The RDF was extracted using STM and for enriching the categories, combining the generated RDF with upper ontologies takes place. Classify the characteristics gleaned from the preprocessed data and enrich categories using the bagging classifier with SVM and Random Forest algorithms. By utilizing the Google KG API to load the knowledge graph from the RDF and compute semantic similarity between categorized entities and knowledge graph entities using the APMI measure. We finalize the indices based on the semantic similarity score.

5 RESULTS AND PERFORMANCE EVALUATION

The performance of the proposed RISL framework, an RDF-driven model for indexing using intelligent semantics and line optimization, is evaluated using Precision, Recall, Accuracy, F-measure percentages, and False Discovery Rate (FDR) as the standard potential metrics. The results' relevance is indicated by their Precision, Recall, Accuracy, and F-measure percentages, while the False Discovery

Rate shows how many false positives the model correctly identified

From Table 1, it is evident that the RISL model yields an overall average Precision of 95.12%, an overall average Recall of 96.27%, an overall average Accuracy of 95.69%, and an overall average F-measure of 95.69% and an FDR of 0.05. To evaluate the performance of the RISL model, it is baselined with WIPOT, IWPSF and MCIR frameworks for indexing. The WIPOT model yields an overall average Precision of 89.22%, overall average Recall of 91.36%, the overall average Accuracy of 90.29%, an overall average F-measure of 90.28% and an FDR of 0.11. The IWPSF model yields an overall average Precision of 90.12%, overall average Recall of 92.09%, an overall average Accuracy of 91.10%, overall average F-measure of 91.09% and an FDR of 0.10. The MCIR model furnishes an overall average Precision of 91.17%, overall average Recall of 93.96%, overall average Accuracy of 92.56%, overall average F-measure of 92.54% and an FDR of 0.09. From Table 1, it is clear that the RISL model yields the highest Precision, Recall, Accuracy, F-measure percentages and the lowest value of FDR compared to the baseline models. The baseline models were evaluated using the same environment and dataset as the proposed RISL model.

The proposed RISL model yields the highest Precision, Recall, Accuracy, F-measure percentages, and the lowest value of FDR because it is driven by three distinct models, namely the TF – IDF, Structural Topic Modeling (STM) and the RDF. RDF generation yields more vital co-occurrent terms for depicting lateral semantics. Apart from Structural Topic Modeling and RDF for generating background knowledge, the upper ontologies also enrich the knowledge into the localized framework. Google's Knowledge Graph API further enhances the density of knowledge in terms of concepts, sub-concepts and instances. So, as a result, the RISL model has a solid knowledge base. In addition, the dataset is classified using the Bagging model with SVC and Random Forest Classifiers as the ensemble. Most importantly, semantic similarity is computed using three strong measures: the adaptive pointwise machine information, Heip's evenness index and Pearson's correlation coefficient. In addition, a line optimization algorithm ensures the transformation of the initial feasible solution set into much more relevant optimization. Henceforth, the proposed arrays and framework are much better than the baseline models.

Table 1. Comparison of performance between the proposed RISL model and other baseline models.

Model	Average Precision %	Average Recall %	Average Accuracy %	Average F-Measure %	FDR
WIPOT [1]	89.22	91.36	90.29	90.28	0.11
IWPSF [2]	90.12	92.09	91.10	91.09	0.10
MCIR [3]	91.17	93.96	92.56	92.54	0.09
Proposed RISL Model	**95.12**	**96.27**	**95.69**	**95.69**	**0.05**

The WIPOT model does not perform as expected because, while ontological terms are used for auxiliary provisioning knowledge, the ontologies are static. The ontologies are diverse but are often shallow, and the relevance computation mechanism is not highly comprehensible or cognizable. Therefore, compared to the proposed RISL model, the WIPOT model lags.The IWPSF model also does not perform as well as the proposed framework because it relies solely on similarity features. There is room for improvement in the similar features used. The features are selected and added to the framework solely by similarity schemes, negating auxiliary knowledge use. IWPSF model underperforms due to its dependence on a low-level dataset for feature selection.The MCIR model also does not perform as predicted because, despite having multi-criteria indexing and ranking model for large-scale data, the ranking and indexing are done using the data. There is no knowledge transformation nor any utilization of existing knowledge. Even though multiple criteria are imbibed onto the extensive data, the data is so large that it fails to discover knowledge from the data. Henceforth, the MCIR does not perform as expected.Since the proposed RISL model is driven by TF – IDF, RDF, upper ontologies, robust relevance computation mechanisms, and strong feature control machine learning bagging classifier, it performs much better than the baseline models.

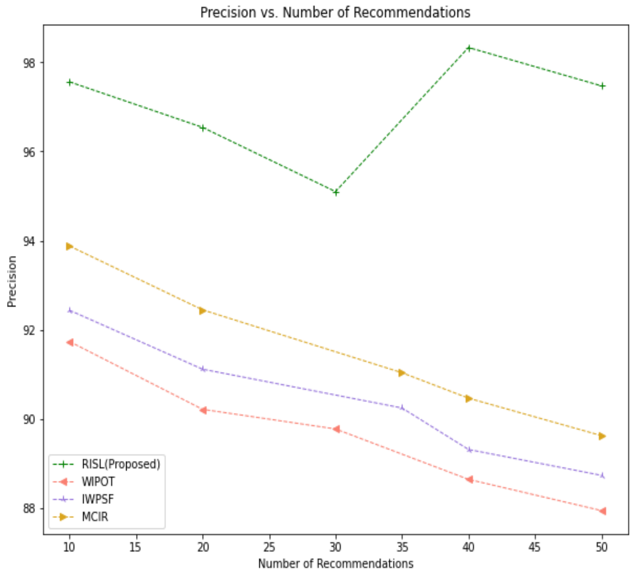

Fig.2. Precision vs Number of Recommendations distribution curve

In Figure 2, the precision vs the number of the recommendations distribution curve is shown. It is indicated that the proposed RISL framework occupies the highest position in the hierarchy. The RISL model is at the top of the hierarchy since it is driven by models such as TF – IDF, Structural Topic Modeling (STM) RDF, and upper ontologies. Strong measures are used to compute the semantic similarity, and the RISL model has a robust knowledge base. The MCIR model comes next in the hierarchy. Since there is no knowledge transformation or use of existing knowledge, it does not perform as expected. The IWPSF model is third in the hierarchy. Its failure to employ auxiliary knowledge and reliance on a low-level dataset for feature selection causes it to perform below expectations. The WIPOT model is listed last in the hierarchy. It performs poorly because it relies on shallow, albeit diversified, ontologies to generate auxiliary knowledge.

7 CONCLUSIONS

Indexing is an essential criterion in today's world due to the overload of information content on Web 3.0. Knowledge-centric Web document tagging has been proposed, which integrates the term frequency-inverse document frequency model with the structural topic modelling to generate the initial topic and the informative word set, which is further used for generating the RDF, which is in turn anchored with relevant upper ontologies to furnish enrich category which is fed as features to classify the data using ensemble bagging classifier. The bagging classifier is a constituent of the independent support vector and the random forest classifier. The auxiliary knowledge is further enriched by loading knowledge graphs using google knowledge graph search API. The relevance computation in the proposed framework is computed using the heap's evenness index and Pearson's correlation coefficient measure, and the APMI measure with the differential thresholds and step deviation measures. Overall Precision, Recall, Accuracy and F-measure 95.12%, 96.27%, 95.69% and 95.69% respectively enriched with the lowest FDR of 0.05 has been achieved by the proposed model, which is the best in the class model compared to the baseline models as it is knowledge centric RDF base as well it encompasses machine intelligence.

REFERENCES

Mukhopadhyay D, Sinha S. Web-page indexing based on the prioritize ontology terms. InWeb searching and mining 2019 (pp. 75-84). Springer, Singapore.

Attia M, Abdel-Fattah MA, Khedr AE. A proposed multi criteria indexing and ranking model for documents and web pages on large scale data. Journal of King Saud University-Computer and Information Sciences. 2021 Nov 2.

Manjula R, Chilambuchelvan A. An efficient approach for indexing web pages using various similarity features. Advances in Natural and Applied Sciences. 2017 Jul 1;11(9):126-34.

Yazdani M, Jolai F. Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. Journal of computational design and engineering. 2016 Jan 1;3(1):24-36.

Gao F, Chen CH, Hsieh JG, Jeng JH. Support Vector Classifier Trained by Gradient Descent. In2021 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD) 2021 Oct 21 (pp. 1-5). IEEE.

Wright L, Fluharty M, Steptoe A, Fancourt D. How Did People Cope During the COVID-19 Pandemic? A Structural Topic Modelling Analysis of Free-Text Data From 11,000 United Kingdom Adults. Frontiers in psychology. 2022 Jan 1;13.

Zhan C, Zheng Y, Zhang H, Wen Q. Random-forest-Bagging broad learning system with applications for COVID-19 pandemic. IEEE Internet of Things Journal. 2021 Mar 17;8(21):15906-18.

Grootendorst M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv preprint arXiv:2203.05794. 2022 Mar 11.

Liu X, Chen H, Xia W. Overview of Named Entity Recognition. Journal of Contemporary Educational Research. 2022 May 30;6(5):65-8.

Kumar P, Kumar R, Gupta GP, Tripathi R. A Distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing. Transactions on Emerging Telecommunications Technologies. 2021 Jun;32(6):e4112.

Van Assche D. Balancing RDF generation from heterogeneous data sources. InSemantic Web, 19th Extended Semantic Web Conference (ESWC 2022), Lecture Notes in Computer Science 2022 (pp. 1-10).

Khodaei A. Combining textual web search with spatial, temporal and social aspects of the web. University of Southern California; 2013.

Alkwai LM, Nelson ML, Weigle MC. Comparing the archival rate of Arabic, English, Danish, and Korean Language web pages. ACM Transactions on Information Systems (TOIS). 2017 Jun 5;36(1):1-34.

Deepak, G., Kumar, N., Bharadwaj, G. V. S. Y., &Santhanavijayan, A. (2019, December). OntoQuest: an ontological strategy for automatic question generation for e-assessment using static and dynamic knowledge. In 2019 Fifteenth International Conference On Information Processing (ICINPRO) (pp. 1-6). IEEE.

Kaushik, I. S., Deepak, G., &Santhanavijayan, A. (2020). QuantQueryEXP: a novel strategic approach for query expansion based on quantum computing principles. Journal of Discrete Mathematical Sciences and Cryptography, 23(2), 573-584.

Hybridised, K. C. N. OntoKnowNHS: Ontology Driven Knowledge Centric Novel Hybridised Semantic Scheme for Image Recommendation Using Knowledge Graph. Knowledge Graphs and Semantic Web, 138.

Yethindra, D. N., & Deepak, G. (2021, September). A semantic approach for fashion recommendation using logistic regression and ontologies. In 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-6). IEEE.

Deepak, G., &Santhanavijayan, A. (2022). UQSCM-RFD: a query–knowledge interfacing approach for diversified query recommendation in semantic search based on river flow dynamics and dynamic user interaction. Neural Computing and Applications, 34(1), 651-675.

Adithya, V., & Deepak, G. (2021, March). OntoReq: an ontology focused collective knowledge approach for requirement traceability modelling. In European, Asian, Middle Eastern, North African Conference on Management & Information Systems (pp. 358-370). Springer, Cham.

Vishal, K., Deepak, G., &Santhanavijayan, A. (2021). An approach for retrieval of text documents by hybridizing structural topic modeling and pointwise mutual information. In Innovations in Electrical and Electronic Engineering (pp. 969-977). Springer, Singapore.