

# An efficient ontology model with query execution for accurate document content extraction

Poluru Eswaraiah, Hussain Syed

School of Computer Science and Engineering, VIT-AP University, Andhrapradesh, India

## Article Info

### Article history:

Received Jul 14, 2022

Revised Oct 4, 2022

Accepted Oct 21, 2022

### Keywords:

Content extraction  
Information retrieval  
Knowledge discovery  
Ontology model  
Query execution

## ABSTRACT

The technique of extracting important documents from massive data collections is known as information retrieval (IR). The dataset provider coupled with the increasing demand for high-quality retrieval results, has resulted in traditional information retrieval approaches being increasingly insufficient to meet the challenge of providing high-quality search results. Research has concentrated on information retrieval and interactive query formation through ontologies in order to overcome these challenges, with a specific emphasis on enhancing the functionality between information and search queries in order to bring the outcome sets closer to the research requirements of users. In the context of document retrieval technologies, it is a process that assists researchers in extracting documents from data collections. It is discussed in this research how to use ontology-based information retrieval approaches and techniques, taking into account the issues of ontology modelling, processing, and the transformation of ontological knowledge into database search queries. In this research work, an efficient optimized ontology model with query execution for content extraction from documents (OOM-QE-CE) is proposed. The existing ontology-to-database transformation and mapping methodologies are also thoroughly examined in terms of data and semantic loss, structural mapping and domain knowledge applicability.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Poluru Eswaraiah

School of Computer Science and Engineering, VIT-AP University

Inavolu, Beside AP Secretariat, Amaravati, Andhrapradesh 522237, India

Email: eswar9490@gmail.com

## 1. INTRODUCTION

A discipline known as ontology-based information extraction is one in which the extraction process is directed by an ontology [1]. The extraction of information is a multi-step process that includes which was before the text into a machine-readable form and establishing heuristics to find the relevant information to also be extracted [2]. For example, a query like "provide me with a list of all papers written by A in which Y is not an author" can't be easily managed to perform using available information extraction techniques [3]. The process of extracting information from text empowers different applications, including question answering systems that can offer more precise answers.

The proliferation of electronic or textual project specifications is a result of an increase complexity of goods and the development pipeline [4], or the increasing popularity of desktop document technologies. New research difficulties and opportunities have arisen as a result of the availability of so many document resources [5]. In order to create more support for design discovery, learning, and reuse, these include enhancing design information retrieval (IR), creating a systematic index for design documents [6], which record engineers' thoughts and reasoning methods for a specific design, is an important issue to address [7]. In order for engineers

to quickly discover the documents they are looking for this representation that should clearly and properly convey the main design concepts and the links between these concepts [8].

It is a hope that domain-specific design ontology and natural language processing can be used to automatically create a structured and semantically-based representation from unstructured designs documents for the purpose of retrieving design data [9]. The document's detected language patterns can be used to deduce the representation's design concepts and relationships [10]. Various concepts and interactions are linked together to build a concept graph [11]. Incorporating these idea graphs creates an application-specific design ontology that represents the organised content of the company's document repository and an automatically filled knowledge base from prior designs. The semantic rules and ontology representation is shown in Figure 1.

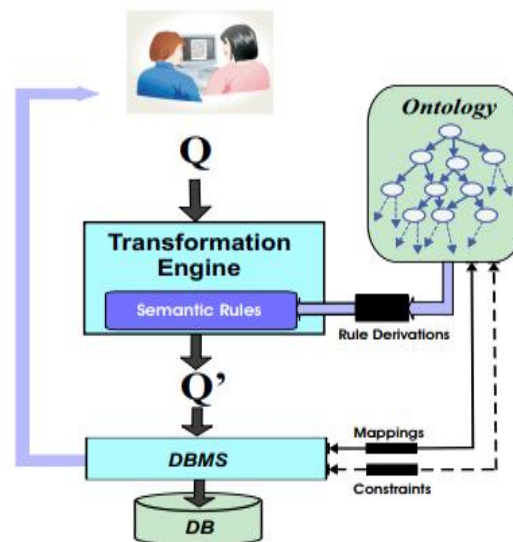


Figure1. Semantic and ontology representation

The obstacles of accessing data via databases and data sources have gotten more sophisticated than they were in the last few decades. In order to solve these issues, we need to hide the data's heterogeneity in format and structure from the consumers [12], as well as overcome the uncertainty in terminologies generated by employing equivalents and homonyms. Consequently, the demand to understand the data from information sources [13] has grown. Syntactic information retrieval is being replaced by semantic information retrieval on google.com. Researchers grow aware of the use of semantic information in dealing with the issues outlined. This meta-data layer can be seen as a semantic knowledge layer over the examples of the original source. It has recently become fashionable to use ontologies to capture this kind of semantics [14]. Because an ontology may give a shared clear agreement of the application in simple and agreeable fashions.

A domain is modelled using terms and relationships provided by ontologies. In order to overcome the interoperability problems of diverse information sources [15], they have shown to be a vital support for data management in databases and information systems [16]. Because of this, users shouldn't give a fig about the structure of the data in the sources [17]. As a result, in systems like OBSERVER [18] and TAMBIS, users can create queries over a particular ontology without having to access the data sources. When it comes to Web search engines, ontologies are also being employed to help improve their functionality [19]. Ontology expertise and inference processes can be annotated onto Web resources to help with search. The semantic web is the culmination of these initiatives and many others. The text extraction process is shown in Figure 2.

An ontology-based technique for improving database query replies is presented in this research. Assuming a database exists, we can infer the presence of an ontology that gives context for the database's items [18]. A more meaningful answer that meets the user's intent can be obtained through the effective usage of an ontology to restate a user query. Selections and projects over database items that meet a set of constraints can be used to define queries [20]. The appropriate response is determined by a collection of terms that are defined. The mismatch between the user's worldview and the database designer's worldview might lead to terms being used by a user that do not perfectly fit the database values while trying to access information about certain items [21]. However, the database may contain values that differ in syntax but are semantically identical to the phrases used by the users and reflect the same purpose. Rather of treating this as a pattern-matching challenge, we treat it as a semantic problem [22].

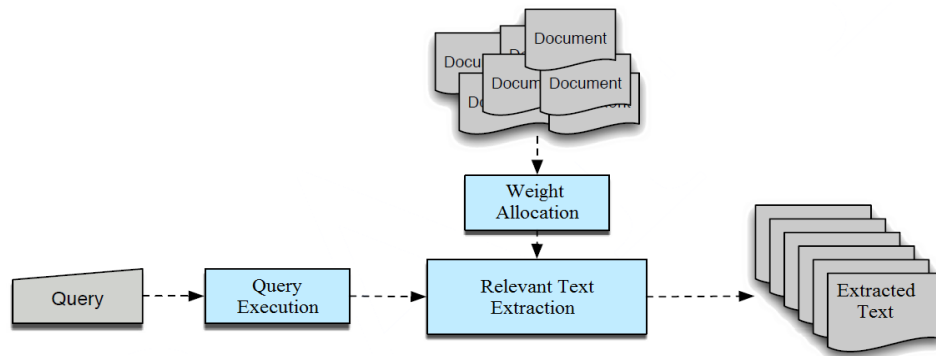


Figure 2. Text extraction

Abstracts and their relationships are used to describe ontologies, which are intended to describe what is known about the essence of entities in a given domain of interest. An ontology's backbone is the hierarchical structuring of concepts via the inheritance relationship [23]. The ontology is sometimes accompanied by a set or logical principles that define the semantics of relationships [24] there are three primary parts to the system. The system's central processing unit, the transformation engine [25], is the initial component. As an input query, it does pre-processing before sending it to the system. This is accomplished by applying a set of lexical rules to remodel  $Q$  into yet another query, such as  $Q$ . Additional semantics gleaned from an ontology underpin these rules [26]. The most important thing the model can do is to expand the scope of user queries by substituting query conditions with alternative conditions that really are semantically identical [27]. Linguistic rules are applied evenly and in any sequence during query reformulation [28].

A reformed question from the previous iteration is utilised to construct a new query iteratively until no further reformulations are possible. If no rules can be implemented to the query, the output query will be the same as the input query [29]. Ontology and databases experts perform the rule derivation procedure by hand. On the basis of information transfers between metaphysical and database entities, a collection of rules is considered. In addition to the underlying database, an ontology is also required. If the database is broad or domain-oriented, the ontology could be one of these two types. The ontology serves as a source of semantic information [30] about the data stored in the database. Finally, the database management systems (DBMS) is responsible for executing the output question and displaying the result to the user. The reformulation rules can be divided into two groups based on augmenting rules and minimization rules. The proposed model develops an efficient ontology model with query execution for accurate document content extraction using natural language processing.

## 2. RELATED WORK

In QUASE model, a question is written as a query, which is typically a noun or verb phrase, rather than a first-order logic expression. QANDA, on the other hand, doesn't use the ontological relationships like QUASE in order to verify responses. Ontology-based information retrieval system [1] Ontoseek is developed [2]. During the retrieval process, it looks for content rather than string. Classified ads and product pages are the primary search targets. An ontology guided graph matching approach is used to solve queries where node stores and edges match if the domain demonstrates that a mathematical relation exists between them. These diagrams were not generated by a computer programme. Through the use of a user-interface, the Ontoseek team has developed a semi-automatic way for verifying linkages between distinct nodes in the generated graph. QUASE is not displaying the query in the form of a graph. Ontoseek does not use machine learning for query classification. In addition, QUASE is retrieving the answer for you.

AQUA is an ontology-driven Question answering system that uses question query language (QLL) to turn natural language questions into logical queries with the help of some pre-defined rules [3]. AQUA uses sentence segmentation and WordNet to help in question classification. Ontology-based queries are accepted by AquaLog, which delivers replies retrieved of one or more knowledge bases (KBs) that fill-up the ontologies given as input by domain-specific knowledge. The NL question is first converted into a query-triple and subsequently into an Onto-triple with the help of the relation similarity service (RSS). An answer engine is also included in RSS, which solves the onto-triple to get the solution [4]. Before it is shown to the user, this response is translated into English using templates. Unlike QUASE, this quaternary ammonium salt (QAS) uses a different approach to question classification. While QUASE does not employ ontologies [5], aqualog leverages operational conceptual simulation environment (OCML) to provide ontology portability [6]. The

DBpedia ontology is utilised for response validation in the QUASE technique, where answers are retrieved from pre-existing documents. Question types can be identified using template mapping techniques, and then a query is routed to a search engine specifically designed to handle them. Their algorithm scours the web for relevant content blocks and considers them to be answers, however they don't supply short responses. QUASE is immune to this problem. Multi-ontology system poweraqua QAS translates natural language queries into RF triples and maps these RF triples [7] across the entire semantic web to obtain an answer.

A multidomain ontology (DBpedia) was utilised to assess candidate responses for QUASE, which relies on question segmentation based on Wierzchon and Klopotek [8] taxonomy. It's looking for answers in predefined natural language material, not the entire web. A named entity is recognised by QAKiS as the proper response to questions comprising a named entity. SPARQL queries can be executed over the metadata SPARQL gateway to retrieve the response to a user's enquiry. In order to classify questions and find relevant documents, it uses machine learning. The DBpedia endpoint is used by QUASE to validate answers.

Syntax analysis and domain knowledge are combined to extract the thoughts from the texts and construct a foundation based on established templates in the context of NLP. Both an expert and a huge training corpus can be used to learn the domain knowledge, which can be structured as expression patterns [10]. In terms of the fundamental techniques and the notion of employing knowledge base to assist extraction [11], IE approaches are indeed the closest for text retrieval. A more comprehensive and systematic representational model is therefore required to aid extraction. The extracted representation models are also used to index documents and assist data retrieval method [12]. Grammar-based parsing, enhanced vector model and classification-based techniques are categorised. By utilising augmented transition networks grammars to derive assembly information from drawing notes for automatic assembly of text [13].

A case-based retrieval method is employed where speedy maintenance is needed. Before the extraction procedure, the technique creates a parsing-tree structure for each record. This strategy relies primarily on knowledge base models that have been developed by the company. For the unstructured design documents, Araújo *et al.* [16] proposed an expanded vector model that is, belief networks. After vector model-based algorithms discovered the significant terms, it was constructed by adding less significant effect but causal correlated terms. It was done using k-means clustering and several other heuristics to establish the causal linkages. The results of the experiment suggest that the idea extension based on the enriched representation model improves query recall. Discriminating phrases can be found in the training dataset, such as the scanned passages of mechanic engineers' handbooks, by using this technique.

To classify the documents, some academics have attempted to develop thesaurus or taxonomies. The dedal project was mentioned in Xu *et al.* [18]. Thesauri, or collections of linked terms, are used to index the electronic design notebooks. Wang *et al.* [20] sought to automate the filling of the dictionary by applying the vector model-based technique and principal component analysis. The documents were indexed using domain taxonomies by Timo and Reisswig [21]. In addition to the available literature and conversations with engineers, the taxonomies were developed. The vector design technique was also utilised to classify the content against the taxonomies. However, the indexing's precision is in doubt. In addition to the full-text search, classification schemas, taxonomies were used to automatically classify diverse technical publications. According to the taxonomy, terms were used to describe both textual material and metadata. Products and task decompositions are instances of words, and document codecs and document types are examples of metadata. An inverted file was created by parsing the documents. "Constraint-based classification," or phrase matching, was used to link the texts to taxonomy words.

### 3. METHOD

By specifying the ideas and interactions between the concepts in a domain ontology, a model of knowledge representation is provided. Using relationships, the ontology model combines several inference techniques into a single framework, making it accessible to humans and machines alike. An ontology model is used in this research as a data structure and model to express the semantics of design. In order to aid in the extraction of semantics from a design document, we use a tiered design ontology model. Domain ontology information can be applied to a broader range of products and organisations when they are all part of the same technical discipline.

Device taxonomies, performance taxonomies, function taxonomies, material and manufacturing processes and environments are all sub ontologies of the domain ontology. Product and component taxonomies are subsets of device taxonomy; property and unit taxonomies are part of performance taxonomy. A separate substructure of the design semantics, each taxonomic is a hierarchical grouping of concepts. The ontology was constructed based on the domain ideas and the subject matter of the domain. The subclasses are built in accordance with the domain notion. Domain concepts would be found at the very heart of an ontology. Annotation ontologies are designed to facilitate semantic indexing.

Structured data is frequently found embedded in text documents. For example, information extraction systems construct structured relations using documents, enabling the processing of expressive, structured queries across text databases. Retrieve relevant textual content, extract relationships from the documents and merge extracted relationships for queries requiring multiple relationships be deconstructed into a set of simple processes. For a variety of reasons, each step in query processing can have a variety of options. As a first step, information extraction methods are not always ideal, and they may provide incorrect data or fail to capture information that they should. Because the output quality of different extraction systems varies, it is possible to use more than an extraction method to uncover a relationship. The total text extraction process represented in Figure 3.

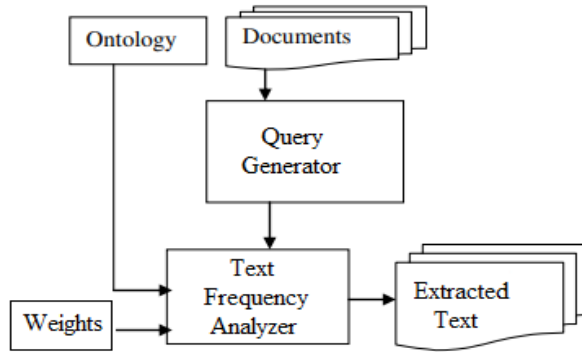


Figure 3. Text extraction process

There is a lot of noise in data extraction, and the relationships that are recovered are neither flawless nor full. Several issues can cause incorrect tuples to be generated by an extraction system. It is also possible that a document’s tuples may not be extracted by an extraction system. An extraction system’s precision and recall, as well as the quantity of good and bad attribute values in the output, can be used to assess the quality of its output. Precision, on the other hand, measures the percentage of excellent tuples the system is able to extract from a text database, while recall represents the percentage of good tuples the system is able to extract. Precision and recall are inherently trade-offs, and extraction systems can be honed to favour either one over the other. In this research work, an efficient optimized ontology model with query execution for content extraction from documents (OOM-QE-CE) is proposed. Weighted values are used to compute ontology annotations in the proposed model, which describe the importance of the documents. The proposed model is explained in the algorithm.

**Algorithm OOM-QE-CE**

{  
 Input: Text Data {TDS} and Ontologies {On}  
 Output: Extracted Text Detection Set  
**Step-1:** Load the document to process and define the predefined ontology set for text extraction model from the input document. The document and record loading is performed as,

$$Record[i](D[M]) = \sum_{i=1}^N \min (getattribute(i) + getvalue(i)\epsilon TDS) \tag{1}$$

Here M is the maximum number of records in the dataset and getattribute () is used to identify the first record in first tuple and the getvalue () is used to extract the value from the first tuple.

**Step-2:** Select the significant query for execution so that text extraction can be performed based on the query provided. The query construction is performed as,

$$Qu[l]^* = \sum_{k,q \in d_i} exec(\sum_{q=1} input(i)) \times \min (len(input(i)))_{i,q} + Th \tag{2}$$

Here a query of length l is considered and Th is threshold value that represents the length of query.

**Step-3:** The ontologies are defined for the document to extract the text from the document based on the input query. The ontology definition process is performed as,

$$Ont(R_{Ns}) = \sum_{i=1} \frac{|record(i) \cap record(i+1)|}{|sizeof(TDS)|} < Th \tag{3}$$

$$FOnt_N \rightarrow Ont(i) + record(Que(i)) \cup getvalue\{TDS(i)\} \tag{4}$$

$$Ruleset[K] = \sum_{i=1}^N \sum_{j \in N_i} (\min(FOnt(i)) + Th) + \min(Ont(i)) \quad (5)$$

**Step-4:** The weight allocation is performed for the ontologies defined for relevant text extraction and to avoid irrelevant text from the documents. The weight allocation is performed as,

$$We_Vec[D] = \{we_{i,D_1}, we_{i,D_2}, \dots, we_{i,D}\} \quad (6)$$

$$WeightSet[FOnt(i)] = \sum_{i=1}^M \frac{We_Vec(i) + \min(We_Vec(i)) + Th}{\max(We_Vec[D])} \quad (7)$$

**Step-5:** The text frequency calculation is performed for identification of the actual and relevant text and the location in the document for extraction. The text frequency is calculated as,

$$fr = \max(\sum_{i \in TDS} \cos(\max(WeightSet(i, i + 1))) + \sum_{i=1}^N \maxcount(getvalue(Record(i)))) \quad (8)$$

**Step-6:** The ranking process is performed to the extracted text for sequence. The text ranking process is used to identify the precision levels. The text ranking is performed as,

$$Ranking(Record(i)) = \max\left(\frac{fr(getvalue(i)) + \max(WeightSet[Record(i)])}{\sum_{i=1} \text{sizeof}(Ruleset(M))}\right) + \max(fr(i, i + 1)) \quad (9)$$

**Step-7:** Display the extracted text set.  
}

#### 4. RESULTS AND DISCUSSION

NLP researchers have recently grown interested in automatic text or document retrieval based on queries. The purpose of this research is to point out the key differences between data retrieval from documents to summarise previous experience in the field and to examine external developments that stimulate growing interest in text and document retrieval, and to take into account appropriate approaches for NLP research work that is aimed at this form of data processing. The proposed model is implemented in Python and executed in Google Colab. The proposed model considers the dataset from the link <https://www.kaggle.com/datasets/rhuebner/human-resources-data-set/>. The proposed optimized ontology model with query execution for content extraction from documents (OOM-QE-CE) is compared with the traditional information extraction based on named entity (IebNE) model in terms of data analysis time levels, query execution accuracy levels, applying ontology model accuracy levels, content extraction time levels, content extraction accuracy levels and error rate.

Data analysis uses machine learning, analytics, and linguistics to identify patterns and trends in unstructured data. Text mining and text analysis, which put the material into a more structured manner, can provide additional quantitative insights. The proposed model analyses the data for performing query execution and extracting relevant information from documents. The proposed model is compared with the traditional model and the results are represented in Figure 4. There are no non-language characteristics, such as the addition symbol or the asterisk, and no special format or altering of grammar in a natural language inquiry, which is what, is meant by a “natural language query.” The query is the general word or a sequence of words provided by the user to extract that information from the document. Based on the query the complete information is retrieved from the document dataset. The proposed model query execution accuracy levels are high than the existing method. The comparison levels among the proposed and traditional models are shown in Figure 5. Additional forms of relationships are usually binary in ontologies. They represent a relation between two notions or entities that are distinct from one another. xRY or predicate form are the most frequent ways to express these relationships. Ontologies are systems for storing and retrieving knowledge that can be applied across different domains. The ability to model high-quality, linked, and coherent data is determined by its ability to express relationships and its high interconnection. The proposed ontology model accuracy levels are contrasted with the existing model and the results represent that the proposed model accuracy levels are high. The comparison levels are shown in Figure 6. The positive results of the proposed model retrieved from document are indicated in Table. The proposed model positive results are high than the traditional models that are shown in Table 1.

To collect or retrieve various sorts of data from many sources, many of which may be badly organised or completely unstructured, data extraction is all about to verify whether the data is retrieved based on the provided query. Emails, web pages, reports, presentations, legal documents, and scientific papers are all examples of textual sources from which information extraction might be used. The content extraction time levels of the proposed and traditional models are shown in Figure 7. The content extraction represents that relevant data is only retrieved from the document of large size that too based on the given query. The content extraction accuracy levels of the proposed and traditional models are shown in Figure 8. The error rate represents the process of retrieving irrelevant data base done the given query. The proposed model and

traditional model error rates are shown in Figure 9. The error rate of the proposed model is very less than the exiting model that indicates the proposed model performance is high.

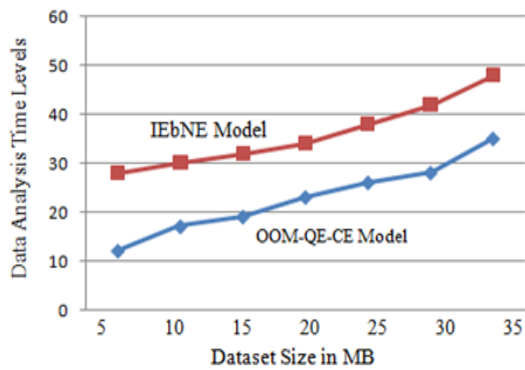


Figure 4. Data analysis time levels

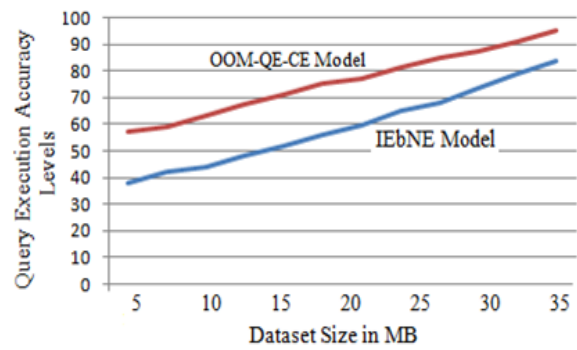


Figure 5. Query execution accuracy levels

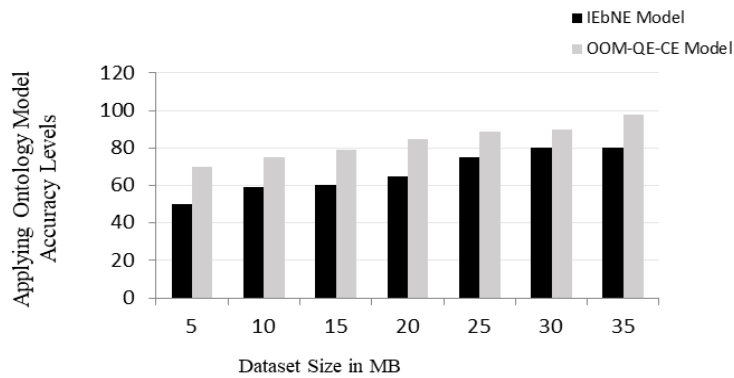


Figure 6. Applying ontology model accuracy levels

**Table 1. Accuracy levels**

Number of Queries	Number of Positive Results	Accuracy
250	235	94%
500	489	98%
750	694	93%
1000	988	99%
1250	1190	95%
1500	1424	95%
1750	1689	97%
2000	1951	98%

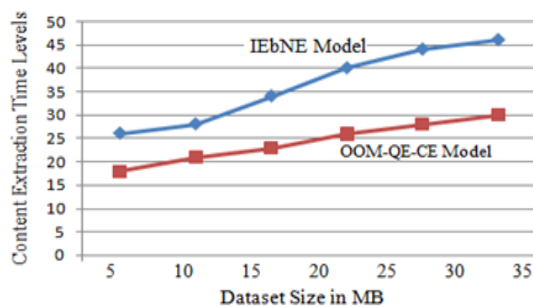


Figure 7. Content extraction time levels

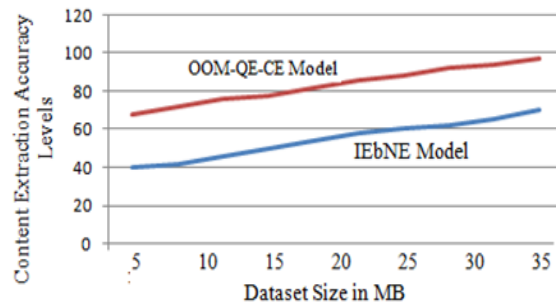


Figure 8. Content extraction accuracy levels

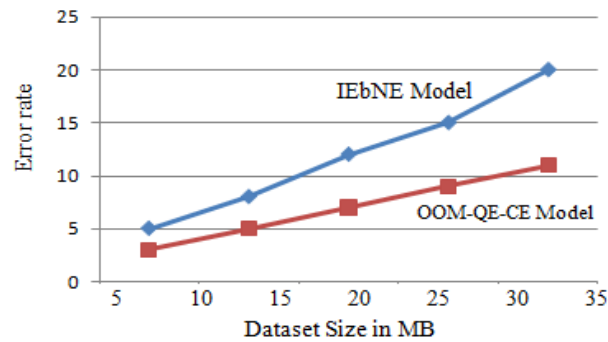


Figure 9. Error rate

## 5. CONCLUSION

Incorporating statistics into the analysis of natural language reflects the traditional paradigm of information retrieval systems and is characterised by the set of key words for each document, known as the index. Based on the bag of words, this is a simple emphasis. All keywords in a text are considered index terms in this method. For each term, its value is assessed by how frequently it appears in the document and how much weight it is given. In the discipline of computer science known as "information retrieval," documents containing free text are processed in order to make them easily searchable using keywords entered by the user. IR technology is critical since it provides the foundation of software that facilitates literature searches on the Web. Words in documents can be indexed, as can ideas that can be matched to domain-specific models, concept matching models that raise various practical issues that make it inappropriate for usage alone. The user queries are pre-processed so that the keywords can be separated. The pre-processing idea is query consolidation, which reduces the number of comparable keywords to one. The scheduling takes into account the needs of the end customers. The proposed scheduler is used to plan the users in accordance with the most important properties. In the proposed research, an efficient optimized ontology model with query execution for content extraction from documents (OOM-QE-CE) is proposed. The proposed model achieves 97% accuracy in text extraction using NLP. The proposed optimized ontology model extracts the accurate content from the documents. In future, the features considered in text extraction can be reduced for reducing the training time and the accuracy rate can be still enhanced and bag of words can be further improved for performance enhancement.

## REFERENCES




- [1] E. Botoeva, D. Calvanese, B. Cogrel, J. Corman, and G. Xiao, "Ontology-based data access—Beyond relational sources," *Intelligenza Artificiale*, vol. 13, no. 1, pp. 21-36, 2019, doi: 10.3233/IA-190023.
- [2] D. Lembo and F. M. Scafoglieri, "A formal framework for coupling document spanners with ontologies," in *In 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2019, doi: 10.1109/AIKE.2019.00036.
- [3] G. D. Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati, "Using ontologies for semantic data integration," in *In A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, Springer, Cham, pp. 187-202, 2018.
- [4] L. Peterfreund, B. Cate, R. Fagin, and B. Kimelfeld, "Recursive programs for document spanners," in *In Proc. 22nd Int. Conf. Database Theory*, 2019, doi: 10.48550/arXiv.1712.08198.
- [5] A. Rezaeipanah, G. Ahmadi, and S. S. Matoori, "A classification approaches to link prediction in multiplex online ego-social networks," *Social Network Analysis and Mining*, vol. 10, no. 1, pp. 1-16, 2020, doi: 10.1007/s13278-020-00639-6.
- [6] B. Selvalakshmi and M. Subramaniam, "Intelligent ontology based semantic information retrieval using feature selection and classification," *Cluster Computing*, vol. 22, no. 5, pp. 12871-12881, 2019, doi: 10.1007/s10586-018-1789-8.
- [7] Z. Alzamil, D. Appelbaum, and R. Nehmer, "An ontological artifact for classifying social media: Text mining analysis for financial data," *International Journal of Accounting Information Systems*, vol. 38, p. 100469, 2020, doi: 10.1016/j.accinf.2020.100469.
- [8] S. T. Wierzchon and M. A. Klopotek, "Modern algorithms of cluster analysis," in *In a Comprehensive Guide Through the Italian Database Research Over the last 25 years*, Springer International Publishing, p. 421, 2018.
- [9] S. Lyu, X. Tian, Y. Li, B. Jiang, and H. Lyu, "Multiclass probabilistic classification vector machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 3906-3919, 2019, doi: 10.1109/TNNLS.2019.2947309.
- [10] H. Zhou, J. Zhang, Y. Zhou, X. Guo, and Y. Ma, "A feature selection algorithm of decision tree based on feature weight," *Expert Systems with Applications*, vol. 164, p. 113842, 2021, doi: 10.1016/j.eswa.2020.113842.
- [11] R. Gupta and T. Rincy, "An efficient feature subset selection approach for machine learning," *Multimedia tools and applications*, vol. 80, no. 8, pp. 12737-12830, 2021, doi: 10.1007/s11042-020-10011-7.
- [12] X. Y. Lu, M. S. Chen, J. L. Wu, P. C. Chang, and M. H. Chen, "A novel ensemble decision tree based on under-sampling and clonal selection for web spam detection," *Pattern Analysis and Applications*, vol. 21, no. 3, pp. 741-754, 2018, doi: 10.1007/s10044-017-0602-2.
- [13] K. Gupta, A. Khajuria, N. Chatterjee, P. Joshi, and D. Joshi, "Rule based classification of neurodegenerative diseases using data driven gait features," *Health and Technology*, vol. 9, no. 4, pp. 547-560, 2019, doi: 10.1007/s12553-018-0274-y.
- [14] R. McDonald, G. I. Brokos, and I. Androutsopoulos, "Deep relevance ranking using enhanced document-query interactions," in *EMNLP, In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, doi: 10.48550/arXiv.1809.01682.






- [15] A. Rehman, K. Javed, and H. A. Babri, "Feature selection based on a normalized difference measure for text classification," *Information Processing and Management*, vol. 53, no. 2, pp. 473-489, 2017, doi: 10.1016/j.ipm.2016.12.004.
- [16] G. Araújo, A. Mourão, and J. Magalhães, "NOVA search at precision medicine 2017," in *In Proceedings of the Twenty-Sixth Text REtrieval Conference (TREC) Proceedings*, Gaithersburg, USA, 2017.
- [17] L. Afuan, A. Ashari, and Y. Suyanto, "A study: query expansion methods in information retrieval," in *In Journal of Physics: Conference Series*, vol. 1367, no. 1, p. 012001, IOP Publishing., 2019, doi: 10.1088/1742-6596/1367/1/012001.
- [18] B. Xu *et al.* "A supervised term ranking model for diversity enhanced biomedical information retrieval," *BMC bioinformatics*, vol. 20, no. 16, pp. 1-1, 2019, doi: 10.1186/s12859-019-3080-2.
- [19] M. Agosti, G. M. Di Nunzio, and S. Marchesin, "An analysis of query reformulation techniques for precision medicine," in *In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, doi: 10.1145/3331184.3331289.
- [20] J. Wang, M. Pan, T. He, X. Huang, X. Wang, and X. Tu, "A pseudo-relevance feedback framework combining relevance matching and semantic matching for information retrieval," *Information Processing and Management*, vol. 57, no. 6, pp. 102342, 2020, doi: 10.1016/j.ipm.2020.102342.
- [21] I. Timo, Denk and C. Reisswig, "BERTgrid: Contextualized embedding for 2d document representation and understanding," in *In Proceedings of the 33rd Conference on Neural Information Processing System*, 2019, doi: 10.48550/arXiv.1909.049484948.
- [22] J. Devlin, M. W. Chang, M. K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistic Human Language Technologies*, 2018, doi: 10.48550/arXiv.1810.04805.
- [23] L. Garncarek, R. Powalski, T. Stanisławek, B. Topolski, P. Halama, M. Turcki, and F. Galiński, "LAMBERT: layout-aware language modeling for information extraction," in *In International Conference on Document Analysis and Recognition*, 2021, doi: 10.1007/978-3-030-86549-8\_34.
- [24] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, and J. B. Faddoul, "Chargrid: Towards understanding 2d documents," in *In Proceedings of the 33rd Conference on Neural Information Processing Systems*, 2018.
- [25] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *In Proceedings of the 9th International Conference on Learning Representation*, 2019.
- [26] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.
- [27] B. P. Majumder, N. Potti, S. Tata, J. B. Wendt, Q. Zhao, and M. Najork, "Representation learning for information extraction from form-like documents," in *In proceedings of the 58th annual meeting of the Association for Computational Linguistic*, pp. 6495-6504, 2020, doi: 10.18653/v1/2020.acl-main.580.
- [28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015, doi: 10.1109/TPAMI.2016.2577031.
- [29] S. Tata, N. Potti, J. B. Wendt, L. B. Costa, M. Najork, and B. Gunel, "Glean: Structured extractions from templatic documents," in *In Proceedings of the Real DB Endowment*, 2021.
- [30] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "Layoutlm: Pre-training of text and layout for document image understanding," in *In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 1192-1200, doi: 10.1145/3394486.3403172.

## BIOGRAPHIES OF AUTHORS



**Poluru Eswaraiah**    was born Ammapalem, Venkatagiri, SPSR Nellore, Andhra Pradesh, India in 1991. He received the B.Tech. and M.Tech. Degree in computer science and engineering from Jawaharlal Nehru Technological University, Anantapur, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the school of computer science and engineering at VIT-AP University, VIT-AP University, Inavolu, Beside AP Secretariat, Amaravati, Andhrapradesh 522237, India. He can be contacted at email: eswar9490@gmail.com.



**Prof Dr. Hussain Syed**    Associate Professor, have 12+ years of professional experience in teaching, research and development including IT-industry experience. He has 16 publications in reputed journals and 4 patents so far. He is currently working as an associate professor in school of computer science and engineering at VIT-AP University, VIT-AP University, Inavolu, Beside AP Secretariat, Amaravati, Andhrapradesh 522237, and India. He can be contacted at email: hussain.syed@vitap.ac.in.