3. ARCHITECTURE OF DOCUMENT BASED SEMANTIC CONTENT MANAGEMENT SYSTEM

3.1 INTRODUCTION

Cloud computing is a current technology which provides ondemand model for all the services provided. There are three service models based on the cloud. Cloud requeries distributed and scalable environment and it also posses a broad network access. Based on the above facts, the content management system is identified as one of the prominent application in SaaS. The cloud-based databases are ever increasing the size of data sets. Since the conventional database services are too much costly for execution, numerous cloud database services and distributed frameworks run in clusters in several computers across the internet. The machine learning techniques like clustering, classification are executed across the clusters to store the documents efficiently. The data in CMS needs semantic search and retrieval which enhances the content probing in the collaborative web. SCMS is designed and works with limited functionalities which are different from the CMS. Therefore there is a need for adding functionalities like the recommendation, ranking and inbuilt change tracking in the existing SCMS.

Surveys of literature carried out in the previous chapter have stressed that there is a need to improve the functionalities of the existing SCMS. CMS is used in most of the professional industries to organize and manage the type of digital information like pictures, audio and video documents, etc. These kinds of systems are featured by the need for adequate storage and reading the massive volume of content. Strict rules are incorporated for controlling and sharing of information among the users and organizations. Nowadays the huge number of web-based CMS are in use. The CMS are featured by storage of pictures, audio and video data. The problem in the web is that it is tough to find relevant information accurately. And also difficult to track the data. To address this issue in semantic web has been proposed as the extension of the current web. There are web-based content management systems and enterprise content management systems which
are implemented in many industries. Adopting the semantics in CMS there is a need to extend the functionality of existing model.

3.2 OVERVIEW OF SEMANTIC CONTENT MANAGEMENT SYSTEM

The traditional CMS is a three-layered architecture and it consists of the following components as shown in Figure 3.1. The user interface, content access, content data model and content repository are the components of the CMS server. The flexibility and adaptability of storage model and repository distinguish CMS for other information systems. The representations of the content are highly flexible to any domain and client scenario. The content management layer provides functionalities for domain-specific data model in CMS.

![Figure 3.1: CMS Server Architecture [71]](image)

Semantic content management system [71] differs from the above traditional CMS regarding architecture and functionality. The content is stored in the repository and additionally processed in the knowledge repository. The content is transferred to knowledge column in SCMS. It is a four layer architecture shown in Figure 3.2 which comprises of presentation & interaction, knowledge management (semantic lifting), knowledge representation & reasoning and knowledge repository.
A knowledge model is created based on the top of the repository. The knowledge repository is generated based on triples of RDF. Semantic user interaction is facilitated with the semantic user interface for identifying the person, thing, an object. Knowledge management is performed with the process called semantic lifting. In this process, knowledge extraction pipelines are designed and the machine learning algorithms are implemented.

This architecture is considered to be primary model for the proposed system. In addition to the functionalities of SCMS, there is a need for additional methods like crawling, ranking, recommendation and tracking of source content. A semantic crawler is necessary for harvesting the URLs and their content. Retrieval of content is enabled with the knowledge model and knowledge access layer. Ranking and recommendation enhance the accuracy of the retrieval. The changes in the sources are tracked with the help of a change tracker. Inbuilt change tracker is added as one of the functionality in the SCMS. The control policies of the access are defined in the knowledge administration for the inbuilt change tracker and DSCMS.

The reference architecture of SCMS can be modified with the enhanced components based on the functionalities. Figure 3.3 shows the proposed modified reference architecture of SCMS. The repository is created.
in a distributed framework and map-reduce paradigms are utilized for processing of content stored. The knowledge model is created based on the semantic analysis of the content and the similarity measures of the entities identified.

![Image: Modified Reference Architecture of SCMS]

Figure 3.3: Modified Reference Architecture of SCMS

### 3.3 PROPOSED DOCUMENT BASED CONTENT MANAGEMENT SYSTEM (DSCMS)

Motivated by the above architectures, a DSCMS model is proposed in this thesis which is confined to a specific domain. The domains include technical documents in computer science, sports, news, business and social media. The CMS is a three-tier architecture model whereas in SCMS the components and functionalities are extended in the same three-layer architecture. The proposed DSCMS model consists of enhanced components in the same three-tier architecture. DSCMS model consists of three phases which embrace the modified reference architecture. The detailed architecture of proposed model is shown in Figure 3.4. It provides a complete end-to-end semantic solution in the distributed framework. The functionality of the SCMS has been enhanced regarding the recommendation, ranking and inbuilt change tracking. The sequence of the process that occurs in DSCMS is described in this section.
The process occurs in three phases such as:

- Semantic Content Marker (SCM) and Semantic Crawler.
- Rendering Engine for Semantic Clustering and Classification (RESCC)
- Hadoop based Semantic Recommendation System (HBSRS)

In the first phase, the semantic user interface and crawler is designed and realized. The proposed system involves two kinds of users namely domain expert users and end users. Initially, if the query from the end user is new, then semantic crawler harvest the URLs and its content from different search engines and websites in a specific domain.

![Figure 3.4: Architecture of DSCMS](image)

It is accomplished by using proposed Semantic Structure and Meta Keyword based Crawler (SSMKC). Crawled links are restricted to weblogs, wikis, official blogs, e-magazines, tutorial sites and question answering
forums. If the query of the end user already exists, then the Semantic Content Marker (SCM) analyses the query and retrieves the documents from the semantic clusters. SCM act as a semantic user interface which implements semantic annotation algorithms and retrieves the relevant websites to users. The storage from SCM is facilitated by different databases and Distributed File System.

The content of the web page, source URL, time stamp and internal links are stored in the repository. To ensure sufficient storage of crawled web pages, pre-processing of web pages is performed. The pre-processor does Pruning, Stemming and Metadata accumulation. The resultant text from the pre-processor is stored in document format in the Distributed File System (DFS). Legacy documents from different databases like RDBMS and NoSQL are also loaded to DFS. When the number of users increases, hundreds and thousands of documents are stored. The second phase involves the storage and categorizing of documents based on machine learning algorithms.

Grouping of documents is essential for efficient searching and retrieval from the colossal repository. To this group, the similar documents semantic clustering algorithms are proposed and implemented in the distributed framework. Semantic clustering is tested initially with preliminary level clustering like semantic K-means and then the quality of clusters are improvised by agglomerative clustering. A Mapper and a Reducer for Semantic Hierarchical Agglomerative Clustering (SHAC) is proposed. SHAC is implemented based on Jaccard similarity and Cosine Similarity and it proved to be the best solution for clustering the documents compared with the conventional similarity measures. It also enables faster searching and retrieval. The clustering results are given as input to classifiers. In the proposed system, two kinds of semantic classifiers developed. Firstly, Semantic score based classifier is implemented to rank the most relevant documents using SDWR. Relevant documents are retrieved based on authoritative score and centrality score using Top-K ranking algorithm. The Top-k ranking is used to retrieve URLs based on based on k – value and priority. The admin user retrieves the intermediate results based on this type
of ranking. Secondly, Sequence-based Naïve Bayesian classifier (SNBC) is developed which identifies sequence words based on word sense in the next noun word in a frame. It resolves the issue of polysemy occurrences of words. The third phase describes the layer of ranking, recommendation and inbuilt change tracker. The documents are ranked based on the Hub score Page Ranking technique (HPR). In recommendation system the filtering is done based on the HPR concept. To recommend the ranked URLs and documents, Hadoop based Semantic Recommendation (HBSRS) system is incorporated in DSCMS. The requirement of HBSRS is to merge clustered web documents based on similarity score and semantic keyword based algorithms to identify the sequence words in the domain. Similarly HBSRS uses collaborative filtering techniques to recommend most relevant URLs. The recommended URLs are converted to Resource Description Framework (RDF) graph sets like subject, predicate, an object. The changes in RDF triples are detected using semantic relationships. Inbuilt change tracker uses a dynamic streaming algorithm and accumulates the changed documents based on the timestamp in the URLs. The dynamic changes in the content of weblogs are monitored and consolidated alerts mailed to the end users periodically with the help of an active streaming algorithm.

Based on the sequential process the implication of semantic analysis for proposed model is described in the next section. Thus the proposed model provides end to end semantic solution for all the components of DSCMS.

3.4 ALGORITHM FOR DSCMS MODEL

The overall process of DSCMS model is facilitated by functionalities of the components. There are two kinds of users who access this system for efficient retrieval and precise searching. The end users retrieve the recommended URLs and receive alerts based on changes in the marked web documents. The usage of semantic analysis is also stressed throughout the algorithm. In this section, the complete procedure of the DSCMS model is given in Algorithm 3.1 which is given in Appendix I.
3.5 SEMANTIC FOURFOLD SIMILARITY MEASURE IN DFS

A fourfold similarity measure is introduced from the initial stage to the final retrieval stage of the model. The semantic similarity measures include domain based similarity with semantic annotation, Cosine / Jaccard similarity based on relevancy score, word sense based similarity and sequence words based similarity approach. Figure 3.5 shows the semantic similarity measures used in DSCMS model. Semantic similarity measure plays a significant role in all the phases of DSCMS model.

The SCM and query parser use semantic annotation technique for parsing the query and matching the source links. Concept matching is used for identifying and locating the linked references in the SCM. Named entity recognition (NER) is utilized for finding the person, object and thing in the user interface. When the user is creating new documents (VD’s) in SCM, the user defines the entity and annotates it for future references. The source web pages are internally linked with the annotated content based on concept matching.

In the rendering engine, the combination of clustering and classification group the similar documents. Semantic similarity is introduced in both the techniques. In the case semantic clustering, the similarity measure is calculated based on domain-specific entities, the title of the documents and high-frequency entities inside the documents. The relevancy among all the three entities is measured concerning the existing domain ontology and statistical methods. The relevancy score is fetched from the generated of XML / RDF file is shown in Figure 3.6. This relevancy score is matched with the help of cosine similarity and Jaccard similarity for evaluation purpose.
While building semantic classifiers, semantic relationship score is considered. The word sense is predicted and the noun, adjective and adverb formats are segregated. Based on the knowledge and quality of the words, meaningful words are detached and the relationship between the concept of the source documents and the matched virtual documents are calculated.

Sequence-based classifiers are constructed based on two or three meaningful consecutive words in a domain. The word sense disambiguation is resolved using this technique. While matching the critical two consecutive words may retrieve documents from both domains for neutral keywords like “computer architecture.” This kind of retrieval reduces the accuracy. Hence to resolve the problem SNB classifiers are proposed in the DSCMS. The recommendation uses sequence words based methods with user choice based algorithms for personalization. Whereas ranking method introduces HPR technique which utilizes the inter and intra ontology concept.
The graph set is generated for a URL contents called triples namely subject, predicate and object. Based on the graph set ABFS is used to identify the same subject with changed literals. The content change is also monitored by comparing the new content semantically with the old content dynamically. The RDF triples and the rule based content matching algorithm results in the changed alerts. It is obvious from the directed graph set, the set of the triples and its entities are identified and saved in the data store.

Parts of speech recognitions, tokenization of words that parsed from the query, number of occurrences, text content of that particular URL are extracted and stored in repository.
The sample ontology and relevancy score between the entities in the range of 0-1 scale is shown in the Figure 3.7. Effective storage, retrieval and tracking are made possible by applying the fourfold semantic similarity measure in various components of DSCMS model as shown in the following and the same is explained in detail in forthcoming sections:

**Semantic Crawler and Semantic Content Marker (SCM):** Semantic Crawler harvests the URLs and its content. It stores the source URL, content and timestamp in databases. If the user and the query are new, then SCM forward the query to crawler for URL harvesting. It also does pre-processing of the extracted content and stores it in distributed framework. If the user and the query already exist, then it analyses the query directly and retrieves the document.

**Semantic Clustering of Documents:** Semantic clustering groups the similar documents and the efficient clustering algorithm are proposed and correspondingly accuracy is improved.
Semantic Similarity Score based Classification: Score based classification is applied based on the threshold values finalized from the semantic clustering methods.

Sequence Words based Classification: Content-based classification is developed to enhance the score based classification. Consecutive two sequence words are identified from parts of speech analysis using WordNet and added to the libraries.

Hadoop Based Semantic Recommendation System (HBSRS): HBSRS is developed and an inbuilt change tracker is incorporated based on semantic relationships in RDF.

3.5.1 Semantic Content Marker and Crawler for data extraction

The URL harvesting is the major work since suitable data is selected as source documents [72]. A web crawler is designed which automatically collects the URLs and sources which matches the input query of the user [73]. The Initial phase is that collection of data stored in the web and crawling of their source content appropriately based on Virtual Document (VD) context. VD is a newly created document by the user from the existing web resources. The structure of the existing blog URL is analyzed before extracting content from web pages. The extraction of the URL is done by matching the URL content with the meta-keyword of the seed URL. The title of the website or blog content is extracted. The crawler is used for extracting links in the blogs such as incoming and outgoing links. The meta-keywords of seed URL are accounted and stored in the database. Seed URLs are identified based on the domain for the crawler. In the case study, 39,000 URLs are crawled in various domains such as technical domain like Computer science, Sports Domain, News and Social media Domain. Out of which 9000 URLs are crawled based on blog structure, domain and semantic relevancy. The crawler is designed to harvest the URLs based on semantic relevancy and meta-keywords from seed URL. Harvest Rate indicates the rate of significant pages crawled and successfully taking out unrelated URLs. The harvested URL and its content are transformed into text documents and
stored in DFS. The similar documents are grouped together and stored in DFS using semantic clustering techniques

3.5.2 Rendering Engine

Semantic clustering algorithm is proposed in the rendering engine component of DSCMS. Rendering engine is one which renders the HTML / XML / RDF pattern to the recommendation engine. This engine is built on DFS which stores the virtual documents and source files through content marker. The source files linked with the virtual documents are also stored in databases such as RDBMS [74], NoSQL. The stored data are loaded to DFS. To group the huge amount of data there is a need for clustering. It aims automatically to divide the web documents into different categories based on the semantic score. Clustering is one of the unsupervised learning and in this research work the output data from these clusters are fed to classification which is a labelled or supervised learning [75]. The semantic K-means clustering and semantic hierarchical agglomerative clustering algorithm is proposed and implemented to improve the efficiency of clustering.

3.5.3 Semantic Similarity Score based Classification and Ranking

The multiple source websites are clustered via proposed rendering system. In order to rank the documents by semantic analysis in hadoop using ontology based SDWR technique is proposed. In the existing dual walk algorithm the scores are calculated based on inter ontology and intra ontology links. It is also based on back links of ontology and reverse of page ranking algorithm. In rendering engine, the RDF / XML documents are stored along with the source documents. The RDF document is given as input to protégé tool and centrality and authoritative scores are generated. The Ontology is visualized in protégé by the RDF document rendered by Alchemy API. Nearly, 5000 documents are tested for the SDWR method and results are plotted which showed better precision than existing ranking techniques.
3.5.4 Sequence Words based Classification

Semantic similarity is the practical, widely used approach to address the natural language understanding issue in many core NLP tasks such as paraphrase identification, Question Answering, Natural Language Generation and Intelligent Tutoring Systems [76]. Word sense disambiguate is used to determine the most appropriate sense for an ambiguous word given a context [77]. Approaches for this work include supervised learning, unsupervised learning and combinations of them. Naive Bayesian classification is benchmarked as best supervised algorithm which can be applied to Word sense disambiguation [78]. Topic models can be considered an advanced model over word distributions: every article is represented by a topic distribution, which in turn is a distribution over words. Similarity between documents can be measured by comparing topic distributions. In our proposed research retrieved recommended sources in content management system (Wikipedia & Weblogs) are implemented by using semantic similarity of source documents and virtual documents using Sequence words based Naïve Bayesian Classification [79] using Bigram method which shows better performance than usual cosine similarity technique.

3.5.5 Recommender System with Inbuilt Semantic Change Tracker

The RDF triples are used to update the changed information in the web document by dynamically monitoring the documents with reduced processing time. Recommendation system based on efficient pruning technique collaborative filtering is proposed. Collection of the documents was extracted from blogs like Tumblr and WordPress. The semantic score cluster identification algorithm is used to group the documents into specific clusters. User focussed blog crawling and collaborative filtering methods are used to recommend the grouped documents. Inbuilt change tracker is developed with the automatic breadth first search in the summarized RDF digraphs [81]. The changes are detected using Flume, streaming tool in Hadoop. The identified changes are mailed to users periodically.
The inbuilt-change tracker detects the changes in the source documents which are stored for future references. The report on changes will updated for a stipulated time, say for example changes every day. The changes can also be triggered by the user. The time take can for detecting and alerting the user is vary less compared to conventional systems. The time reduction is due to updating RDF triples with a streamer.

If the change occurs, when the client transmits the event to its first-hop destination, it will be reported back to the client which allows the application to generate events to take appropriate action.

3.6 SEMANTIC SIMILARITY MEASURE USING NLP TOOLS

In the four-fold similarity, the Semantic Annotation, NER based recognition is based on Alchemy API, whereas the Concept matching and Word Sense Disambiguation (WSD) based sequence word similarity is based on Word-Net. Semantic similarity is a metric defined over a set of documents or terms, where the idea of the similarity is that the relation between the word or concepts is identified by the distance measure calculated by machine learning algorithms and statistical methods. The similarity measuring tools are instead a mathematical tool which calculates the weight of the semantic relationship. These tools calculate the weight of the semantic relationship between units of language, concepts or instances, through a mathematical description obtained according to the comparison of information supporting their meaning or describing their nature. The semantic similarity and semantic relatedness are two different computations where similarity is calculated by "is a" relation. Semantic relatedness is done by the relation between the terms. Semantic similarity is calculated based on the distance between the two ontological concepts. Concept matching is a simple measure for partially listed sets depicted as nodes of a digraph. It would be the shortest-path linking between the two concept nodes. Based on text analysis, semantic relatedness between units of language (e.g., words, sentences) can also be estimated using statistical means of vector space model.
The components of DSCMS such as SCM uses semantic annotation and Named Entity based recognition, whereas SSKM Crawler, Semantic Clustering uses concept matching, which is computed from Alchemy API. Semantic classification uses ontology, sequence words similarity and updating of triples in RDF file are computed based on WordNet and text razor.

The semantic similarity is measured using NLP libraries based on Alchemy API. The relationship between the terms is determined by NLTK libraries based on the WordNet semantic resource. In the proposed system, the libraries and algorithms are customized for handling the query based scenarios.

Four types of query are handled in the application in chapter 7 using proposed DSCMS system namely are:

- Context-based query
- Word sense based query
- Concept-based query (Anti-frequency queries)
- Sequence words based queries

These four types of queries are handled in the proposed thesis. The solution for queries is endorsed with graphical analysis. The comparison of the resolution for these scenario based queries is elaborated in the subsequent chapters. The advantage of the system is that it provides end to end semantic analysis and processing which combines both concept based and sense based analysis. Combination of relevancy of concept and relationship between the words are carried over very swiftly using Distributed File system (DFS).
3.7 SUMMARY

- In this chapter, the architecture of CMS and SCMS are compared. The DSCMS model is designed with enhanced functionalities.
- Despite the existing approaches in CMS, DSCMS comprises of components such as a semantic content marker, rendering and recommendation engine in a distributed framework and inbuilt change tracker.
- The semantic similarity analysis is acquainted with the techniques provided by the existing NLP tools which are influenced by either by ontological relationships or by similarity-based lexical analysis and concept matching.
- The significance of fourfold similarity measure is elaborated in all the components of the proposed DSCMS model. The impact of the fourfold similarity measure is in each component is exemplified.
- The overall working of the model is illustrated with the help of algorithm which depicts the functions and procedures of each component.
- The research objectives and contributions made by thesis are elaborately illustrated. The components of DSCMS model like SCM, Semantic Crawler, RESCC and HBSRS are well defined with its functionalities.

In the next chapter, the design and implementation of SCM and proposed semantic crawler are illustrated.