Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2022.026095
Article

Tech Science Press

Shallow Neural Network and Ontology-Based Novel Semantic Document Indexing for Information Retrieval

Anil Sharma[1,*] and Suresh Kumar[2]

[1]University School of Information, Communication & Technology, Guru Gobind Singh Indraprastha University, Delhi, 110078, India
[2]Department of Computer Science and Engineering, Netaji Subhas University of Technology, Delhi, 110078, India
*Corresponding Author: Anil Sharma. Email: anilsharma@aiactr.ac.in
Received: 15 December 2021; Accepted: 15 February 2022

Abstract: Information Retrieval (IR) systems are developed to fetch the most relevant content matching the user's information needs from a pool of information. A user expects to get IR results based on the conceptual contents of the query rather than keywords. But traditional IR approaches index documents based on the terms that they contain and ignore semantic descriptions of document contents. This results in a vocabulary gap when queries and documents use different terms to describe the same concept. As a solution to this problem and to improve the performance of IR systems, we have designed a Shallow Neural Network and ontology-based novel approach for semantic document indexing (SNNOntoSDI). The SNNOntoSDI approach identifies the concepts representing a document using the word2vec model (a Shallow Neural Network) and domain ontology. The relevance of a concept in the document is measured by assigning weight to the concept based on its statistical, semantic, and scientific Named Entity features. The parameters of these feature weights are calculated using the Analytic Hierarchy Process (AHP). Finally, concepts are ranked in order of relevance. To empirically evaluate the SNNOntoSDI approach, a series of experiments were carried out on five standard publicly available datasets. The results of experiments demonstrate that the SNNOntoSDI approach outperformed state-of-the-art methods, with an average improvement of 29% and 25% in average accuracy and F-measure respectively.

Keywords: Document indexing; shallow neural network; information retrieval; computer science ontology; concept extraction; natural language processing; semantic web

1 Introduction

Information Retrieval (IR) systems are intended to return the most relevant results matching the user's information needs from a pool of information [1]. In IR, document indexing step identifies keywords that represent the document contents and are used in the retrieval phase to facilitate query/document match [2]. In the literature, many studies based on statistical, semantic, and probabilistic approaches have been

proposed to enhance the performance of IR tasks [3–6]. These methods follow common steps for indexing. First, document preprocessing step, involves tokenization, removal of unwanted features (symbols, special characters, stop words, punctuations), lemmatization, and/or stemming. Lemmatization converts a word to its canonical form, while stemming tends to transform the word into its root. Second, extracting promising terms based on the free or controlled vocabulary approach of document indexing. Third, assigning weights to each promising term extracted from the document. And finally, selecting terms in order of relevance based on term weight.

Traditional IR approaches index documents based on terms contained in them rather than concepts that represent them. This creates a vocabulary gap when queries and documents use different terms to describe the same concept. These IR systems do not include semantically related documents (with no lexical overlap between the query and document terms) in the result set. The use of domain knowledge in document representation has also been successful in capturing term semantics and their relationships, which is a fundamental requirement for an IR task [7,8]. Ontologies have long been employed in every field to formally represent and reason over domain knowledge. Ontology-based methods for document indexing include term semantics and eliminate the problem of vocabulary gap. But they are highly dependent on the degree of coverage and detail of the unique input ontology. Recently, Neural Network models have shown state-of-the-art results in various Natural Language Processing tasks, making them a potential approach for document indexing in the IR field [9–12]. These methods exploit semantic knowledge between words in co-occurring contexts but disregard their valuable semantic relational structures, which is preserved in knowledge bases such as Knowledge Graphs, ontologies and semantic lexicons.

To fill this gap, we have designed a novel Shallow Neural Network, Natural Language Processing and ontology-based document indexing (SNNOntoSDI) approach for IR. The SNNOntoSDI approach identifies the concepts representing a document using a shallow neural network (word2vec model) and Computer Science Ontology (CSO). The proposed approach identifies promising terms from the document text using Natural Language Processing (NLP) methods and provides them as input to the word2vec model. The word2vec model contributes semantically similar terms to document vocabulary, which are mapped onto domain ontology to identify concepts. The SNNOntoSDI approach eradicates the problem of vocabulary gap by performing mapping between document vocabulary and ontology concepts. It extracts the concepts and their semantically related concepts by exploiting CSO. The weights are assigned to the concept based on its statistical, semantic, and scientific Named Entity features in the document. The Multi Criteria Decision-Process (MCDP) was used for calculating parameters for feature weights. The concepts are ranked in order of their feature weights.

This paper is organized as follows: In Section 2, we present related work. Section 3 sheds light on the motivation and objectives of the proposed approach. Section 4 illustrates the proposed semantic document indexing approach. Section 5 describes the experimental setup. In Section 6, results and discussion are presented. In the last section, we concluded the work and presented some future directions.

2 Literature Work

In the literature, many document indexing schemes have been proposed, each with their benefits and limitations [13,14]. These document indexing approaches are categorized into free vocabulary and controlled vocabulary-based schemes.

2.1 Indexing Scheme Based on Free Language

These indexing schemes represent documents with the keywords present in them. Esteva et al. [15] presented a Deep Learning (DL) and Term Frequency (TF)-based multistage model for searching for Covid-19 literature. The researchers proposed a document retriever module that combines sentence

Bidirectional Encoder Representations from Transformers (BERT) with keyword-based (Term Frequency Inverse Document Frequency (TF-IDF), Best Match 25 (BM25)) methods to fetch the top 1000 documents matching the user's query and re-rank documents based on relevance weight from two modules, namely Question-Answering and Abstractive Summarization.

Ebadi et al. [16] explored a memory network-based IR system for the recognition of misinformation. This system works in two stages: first, it discovered claim-article pairs based on TF-IDF and BM25 ranking models. Second, it uses a memory network, a supervised DL model for stance detection. It computes claim-article relevance using Cosine similarity. Further, less relevant articles matched by claim-article pairs were filtered out to make the model computationally efficient. Mahalakshmi et al. [17] proposed an image and text retrieval model based on DL. For the image retrieval system, the researchers suggested Convolution Neural Network-based feature extraction, while for the text retrieval system, they utilized Bidirectional Long Short-Term Memory (BiLSTM). For query/image matching, Euclidean distance based similarity was employed.

Gupta et al. [18] introduced DL and Hidden Markov Model-based IR method for retrieving spoken documents. The DL-based Kaldi toolkit was applied to convert audio files to textual scripts. The authors created wavelet tree based indexing of text documents. Finally, the TF-IDF method was employed to create word vectors for documents and query words, while the Cosine similarity method was used for query/document matching.

Wagenpfeil et al. [19] proposed Machine Learning (ML) and Artificial Intelligence (AI)-based generic framework for indexing and retrieving multimedia on smart phones. They began with semantic analysis of multimedia to create a multimedia feature vector graph. Semantic indexing using refined feature weights was created. In their proposal, the authors employed natural languages for query processing and SPARQL Protocol and RDF Query Language (SPARQL) for query representation. Djenouri et al. [20] introduced an IR system employing clustering (Density-Based Spatial Clustering of Applications with Noise (DBSCAN), spectral, k-means) and pattern mining (high utility and frequent pattern mining) schemes. For the ranking of cluster items, two schemes were used. First, query and cluster weight-based pattern score computing; second, query and cluster relevant term-based weighted terms in clusters.

Bhopale et al. [21] proposed an IR system based on Bio-inspired Clustering and Frequent Pattern Mining techniques. The basic idea is to make a document cluster using K-flock clustering and extract patterns using the Recursive Elimination technique. Frequent patterns from queries and documents are matched using the Cosine similarity function. Sharma et al. [22] introduced a NLP-based key-phrase identification and document indexing scheme. This model works on an unsupervised key-phrase extraction scheme based on TF-IDF, word embeddings, phrase-based and external knowledge source-based features. For proposed feature clustering, the Euclidean distance was employed, whereas for word embedding vectors, the Cosine distance was used. The ranking of key phrases is also calculated based on these two scores.

Qiu et al. [23] introduced the Fuzzy set theory and word embedding based IR model. The basic idea is to capture relevant features of the user's information needs and document words using the Continuous Bag of Words (CBOW) approach. The query word and document term's similarity were evaluated on the symmetry property. Dai et al. [24] proposed a contextual neural language modeling-based IR system. The idea behind this proposal was to use ML based BERT word embedding for a deeper understanding of document text and queries in natural language.

2.2 Indexing Scheme Based on Controlled Language

These indexing schemes represent documents with the concepts contained in them, augmented with concepts inferred from external knowledge resources for document text. Zouaoui et al. [25] have

presented an ontology-based search system to extract verses of the Quran. The basic notion is to create an ontology from documents containing text from the Quran. Then the user's query will be converted into SPARQL to search for relevant verses from that ontology. Subramaniam et al. [26] put forward an approach that combines external knowledge sources with topic modeling for IR. The authors utilized a modified Firefly algorithm for selecting document features and Fuzzy c-means clustering for creating document clusters. The Latent Dirichlet Allocation (LDA) was used to retrieve relevant documents.

Boukhari et al. [27] explored a hybrid document indexing scheme based on Description Logic and Vector Space Model (VSM) for the medical domain. The VSM provided an approximate match between the query term and the medical thesaurus. Description Logic provided better presentation of the domain knowledge and inference capability to the proposed scheme. In the end, less relevant concepts were filtered out. Rahimi et al. [28] proposed translation knowledge based cross-language IR. They utilized the aggregation function for frequency and discrimination values of query terms. Document ranking was based on the hierarchical calculation of discrimination values and axiomatic analysis of constraints. Jiang [29] has introduced a multiple knowledge sources based IR model. His proposal exploited knowledge from Wikipedia, WordNet, and domain ontology to model the concept score of a keyword in a document. The researcher introduced the notion of a weighted dynamic semantic network and calculated term weight based on this semantic network using the proposed semantic similarity metric.

Tang et al. [30] integrated ontology and VSM to propose a hybrid semantic IR model. Text and query keyword weights were calculated by employing an ontology based similarity measure. Yu [31] explored the application of query expansion in semantic IR. After preprocessing of documents, important terms from documents were identified and assigned weights according to relevance using Term Frequency. Thereafter, document terms are mapped to concepts in the domain ontology. The Genetic Algorithm was employed to find optimum weight values for each document term. Ontology-based similarity measures were used for query/document matching.

3 Motivation and Objectives

The proposed approach combines two unsupervised techniques, Shallow Neural Networks (word2vec) and external knowledge resource (Computer Science Ontology), to learn the concepts from unstructured text documents. Both techniques are based on learning semantic features from unstructured text documents. A Shallow Neural Network uses a large corpus to learn semantic features among co-occurring terms but disregards semantic relational structures between them [32,33]. This gap is filled by the use of ontologies in the concept extraction task. These semantic relational structures present in the domain ontology enhance our understanding of the relations between two terms/concepts, which increases the accuracy of concept extraction. In addition, our approach is based on approximate matching by utilizing domain ontology, which makes morphological variants of concepts be recognized effectively. The contributions of our proposed approach are as follows:

1. We proposed a methodical framework to integrate a Shallow Neural Network and a domain ontology for semantic document indexing.
2. The SNNOntoSDI approach is capable of extracting concepts even if different terms are used in documents and domain ontology to refer to the same concept. It also provides a sufficient number of semantically similar concepts when the document term has limited or no related concepts present in the domain ontology.
3. We have assigned weights to extracted concepts using its statistical, semantic, and scientific Named Entity features to ensure that the most relevant concepts are ranked higher in the index.

4. For vector representation of document text, we exploited the Skip-gram model trained on 4.65 million scientific publications from Microsoft Academic Graph (MAG) in the computer science domain.
5. The performance of the SNNOntoSDI model is endorsed by comparing it with two state-of-the-art methods on five benchmark datasets in the IR field.

4 Shallow Neural Network and Ontology-Based Novel Semantic Document Indexing

This section presents a Shallow Neural Network and ontology-based novel semantic document indexing approach, which integrates the capability of Neural Network based word embedding and external knowledge resources in solving IR problem. In our proposal, we employed Skip-gram with negative sampling based neural word embedding model and Computer Science Ontology (CSO) as an external resource to extract the key concepts from documents. The first step in the proposed approach is preprocessing of documents, followed by concept identification from domain ontology. The SNNOntoSDI approach is based on a partial match between document vocabulary and external resource terms to accommodate morphological variants by using a string similarity metric.

Candidate concepts are assigned weights based on three features, namely: statistical knowledge, semantic knowledge, and scientific Named Entity knowledge. The concept extraction step may identify a large number of concepts for each document due to partial match. The concept selection step eliminates the less important concepts extracted in the concept extraction step and ranks them based on relevance weight. Fig. 1 illustrates the SNNOntoSDI approach as discussed in the following sub-sections.

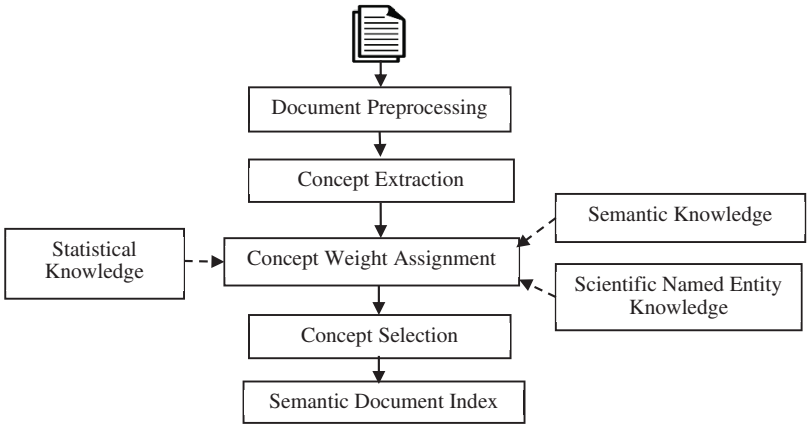

Figure 1: Shallow neural network and ontology-based approach for semantic document indexing

4.1 Document Preprocessing

The document preprocessing step involves converting text to lower case, tokenization, removing punctuations, symbols, special characters, hyperlinks, Hyper Text Markup Language (HTML) tags, and stop words. Stop words are the most common words which occur in text frequently and do not possess any significance for text representation. We also applied lemmatization/stemming and part of speech (POS) tagging using NLP techniques [34].

4.2 Concept Extraction

The quality of the document indexing approach depends on the relevance of concepts representing document text. Concepts in domain ontology are referred to by *n-grams* (unigrams, bigrams, trigrams, and so on) [35]. The architecture of our approach to extracting candidate concepts is depicted in Fig. 2. This part illustrates the concept extraction step of the SNNOntoSDI approach based on [36]. It includes two modules: the syntactic and semantic module. To identify concepts explicitly mentioned in a document, we used the syntactic module. The syntactic module takes the preprocessed text and extracts domain ontology concepts that are directly present in the document. The chunks of text in predefined lexico-syntactic patterns are recognized to identify key phrases by semantic module. This module employs Skip-gram with negative sampling (SGNS) based neural word embedding to get semantically related terms to these key phrases. These semantically similar terms are mapped onto ontology to get concepts and their hypernyms. Finally, we combined the results from these two modules to get an exhaustive list of concepts for a document.

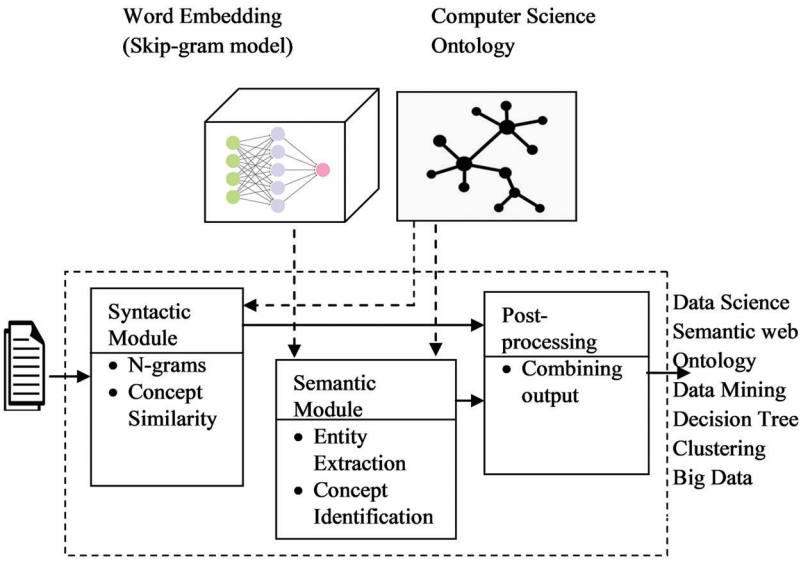

Figure 2: Concept extraction using Skip-gram model and domain ontology

4.2.1 Syntactic Module

The syntactic module creates a mapping between *n-grams* from the documents and concepts in the ontology. Initially, NLP preprocessing techniques are applied to the document text and *n-grams* were created. For each *n-gram*, it calculates the *Levenshtein similarity* with the ontology concept. The ontology concepts with similarity equal or higher than the threshold *mst* were chosen for the list of extracted concepts. Algo. 1 demonstrates the working of this module.

The value for *mst* was empirically set at 0.94. A suitable value of *mst* is capable of handling many variations of ontology concepts, plurals, and hyphens between words. For example, “heuristic approach” and “heuristic-approach”, “heuristic approach” and “heuristic approaches”.

Algorithm 1: Concept extraction using the syntactic module

Input: Document *D*, Computer Science Ontology *cso*, minimum similarity threshold *mst*

Output: List of concepts *concepts*

```
D = TextPreProcessing(D);          /* tokenization, stop word removal */
nGrams = { };                      // empty set of n-grams
for i = 1 to 3 do                  /* Collecting unigram, bigram, trigram */
  nGrams.append(GetNGrams(D));
end for
for each ngram in nGrams do        /* Extracting concepts from CSO */
 for each concept in cso do
   similarity = StringSimilarity(ngram, concept, 'levenshtein');
   if similarity ≥ mst then
     concepts.append(concept);
   end if
 end for
end for
return(concepts);
```

4.2.2 Semantic Module

This module identifies concepts semantically related to documents but may not be mentioned in it. We apply SGNS based word embedding to calculate semantic similarity between document *n-grams* and ontology entities. The concept representation follows a definite lexico-syntactic pattern, such as an adjective or noun followed by one or more nouns. Focusing on chunks following these patterns makes computations fast and avoids combinations that may produce false positive concepts. A grammar-based chunk parser was applied to POS-tagged tokens to recognize these chunks, using the grammar shown in Eq. (1).

$$<JJ.*>*<NN.*>+ \tag{1}$$

where, *JJ* and *NN* indicate adjectives and nouns respectively. These identified chunks are converted into unigrams, bigrams, and trigrams to get their semantically related words from SGNS model using Cosine similarity equal to or higher than 0.7. The space between bigram and trigram token words is replaced by an 'underscore' symbol to identify the corresponding word in the SGNS model. If any bigram or trigram token is not found in the vocabulary of the SGNS model, the average of embedded vectors for all its words is utilized. Then we map all the *n-grams* and their similar words to ontology labels to extract concepts. A list of ontology concepts matching these *n-grams* is created and enhanced by including concepts from hypernymy relations from CSO [37]. Finally, we combined the outputs from both modules. Algo. 2 illustrates the working of the semantic module.

Algorithm 2: Concept extraction using the semantic module

Input: Document *D*, Computer Science Ontology *cso*, Skip-gram with negative sampling *model*, minimum similarity threshold *mst*

Output: List of concepts *concepts*

```
D = TextPreProcessing(D);        /* tokenization, stop word removal */
pos_tags = part_of_speech_tagging(D); // POS tagging
chunks = grammer_parser(pos_tags, '<JJ.*>*<NN.*>+');
for each chunk in chunks do
 ngrams = create_ngrams(chunk, 1, 3); // creating unigrams, bigrams, trigrams
 for each ngram in ngrams do
  gram = join(ngram, '_');
  if gram in model then
    sim_words =model.similar(gram, topn = 10, min_cosine_sim = 0.7);
  else          // in case whole ngram is not in the model
    embedd = [ ];
    for each word in ngram do
      embedd.append(model[word]);
    end for
    ng_embed = mean(embedd);
    sim_words = model.similar(ng_embed, topn = 10, min_cosine_sim = 0.7);
  end if
  for each word in sim_words do        /* Extracting concepts from CSO */
    for each concept in cso do
      similarity = StringSimilarity(word, concept, 'levenshtein');
      if similarity ≥ mst then
        concept + = getSuperConceptOf(concept);
        concepts.append(concept);
      end if
    end for
  end for
 end for
end for
return(concepts);
```

Sometimes an ontology concept is referred to multiple times if multiple *n-grams* may relate to the same ontology concept or the same *n-gram* occurs multiple times in document text. E.g. consider the following *n-grams*: "word vectorization", "word2vec", "Glove", "FastText", which all refer to the same concept: "word embedding models". For this reason, weights are assigned to the concepts based on their overall relevance to the document. The relevance of a concept is calculated using Eq. (3).

4.3 Concept Weight Assignment

In our proposal, the relevance of a concept to a document is modeled by assigning weights to the concept based on three features: statistical knowledge, semantic knowledge and scientific Named Entity knowledge. We employed TF-IDF method to estimate statistical knowledge of a concept [38]. Although TF-IDF is a successful document representation method, but failed to capture the semantic context of the document's text. Ontology-based methods deal with the semantic context of a document by representing domain knowledge inferred from text [32]. The concept of Named Entity recognition has also been successful in enhancing the IR system's accuracy [39]. We have linearly combined the weights from these three features to get the final concept weight. The weight w of any concept c in *corpus* D is assigned as follows:

$$w(c,\ D) = \sum_{d \in D} w(c,\ d) \tag{2}$$

The weight w of a concept c in document d is given by:

$$w(c,\ d) = \ \alpha\ Stat_{score}(c,\ d) + \ \ \beta\ Sem_{score}(c,\ d) + \gamma\ SciNE_{score}(c) \tag{3}$$

where the parameters α, β and γ are weights of features, and these weights are defined according to the problem. The procedure for calculating these weights is discussed in the next section. The term $Stat_{score}(c,\ d)$ represents statistical knowledge about the concept c in document d and is computed as follows:

$$Stat_{score}(c,\ d) = TF - IDF\ (c,\ d) \tag{4}$$

where, TF-IDF is Term Frequency Inverse Document Frequency. The term $Sem_{score}(c,\ d)$ represents semantic knowledge about the concept c in document d and is calculated as follows [36]:

$$Sem_{Score}(c,\ d) = \begin{pmatrix} Number\ of\ times\ a \\ concept\ is\ identified \\ in\ Ontology \end{pmatrix} * \begin{pmatrix} Number\ of\ unique \\ ngrams\ identifying \\ this\ concept\ for\ 'd' \end{pmatrix} \tag{5}$$

where the term $SciNE_{score}(c)$ represents scientific Named Entity knowledge about the concept c and is measured by the fact that the candidate concept belongs to scientific Named Entity for the domain or not [40]. If concept c is present in the scientific Named Entity category of the domain, we assign score of 0.5; otherwise, score 0 is assigned.

$$SciNE_{score}(c) = \begin{cases} 0.5\ ;\ c \in Named\_Entity\_Category \\ 0;\quad Otherwise \end{cases} \tag{6}$$

4.4 Concept Selection

The concept extraction step may identify a large number of concepts, some of which are only marginally related to the document. For this reason, weights are assigned to the concepts to estimate their relevance to the document. The weight of a concept is calculated using Eq. (3). If a concept is explicitly referred to in the document, its weight is set to the maximum concept weight. In the end, the system provides a relevance/weight-based ranking of the extracted concepts. Finally, for each document, concepts are selected based on their relevance weight, and top-n concepts are returned to create a final index. The steps for the proposed SNNOntoSDI framework are demonstrated in Algo. 3.

Algorithm 3: Shallow Neural Network and ontology-based semantic document indexing

Input: *Corpus corpus*, Computer Science Ontology *cso*, minimum weight threshold *mwt*

Output: Semantic document index *SemDocIndex* corresponding to documents in corpus

```
1. for each D in corpus do
2.   concepts = ConceptExtraction_SyntacticModule;          //use Algo. 1
3.   concepts + = ConceptExtraction_SemanticModule;         //use Algo. 2
4.   for each concept in concepts do
5.    Compute concept_Statistical_Knowledge;               //use Eq. (4)
6.    Compute concept_Semantic_Knowledge;                  //use Eq. (5)
7.    Compute concept_ScientificNE_Knowledge;              //use Eq. (6)
8.    Compute concept_weight; //use Eq. (3)
9.    if concept_weight(concept, D) ≥ mwt then
10.      concepts.append(concept);
11.   end if
12.  end for
13.  SemDocIndex = SortAndSelect(concepts, concept_weight);
14. end for
15. Return (SemDocIndex);
```

5 Experimental Setup

The proposed system is implemented using Python 3.9 programming language.

5.1 Description of Datasets

For empirical evaluation of the SNNOntoSDI approach, a set of documents related to scholarly publications in the computer science and library science domains have been chosen and performance was evaluated on standard metrics used in the IR domain. Here, we resorted to five benchmark datasets: Communications of the Association for Computing Machinery (CACM) [41], the Centre for Inventions and Scientific Information (CISI) [42], Library and Information Science Abstracts (LISA) [43], Knowledge Discovery and Data Mining (KDD) [44] and Inspec [45].

The CACM dataset is a collection of 3204 scholarly publications with titles, abstracts and author's details from the CACM journal. The CISI dataset, collected by the Centre for Inventions and Scientific Information (CISI), University of Glasgow consists of 1,460 scholarly publications from the library science domain. The LISA dataset is collected by the Information Retrieval Group at the University of Glasgow and contains 6004 documents containing library and information science abstracts. The KDD dataset consists of 834 abstracts on knowledge discovery and data mining collected from Association for Computing Machinery (ACM) conferences. The Inspec dataset, created by the Institution of Engineering and Technology (IET), contains over 15 million abstracts and indexing records. We considered a collection of 500 abstracts of scientific publications in the fields of computer & control and IT (Information Technology) from Inspec. Each dataset is provided with a set of relevant concepts assigned by domain experts. Tab. 1 provides more details about the valuable statistics available with each dataset.

Table 1: Statistics of datasets used in proposed work

Dataset	Document type	# Documents	# Words
CACM	Computer Science	3204	368460
CISI	Library Science	1460	166440
LISA	Library and Information Science	6004	360240
KDD	Computer Science	834	83230
Inspec	Computers and Controls, IT	500	60110

5.2 Domain Ontology

The SNNOntoSDI approach utilized Computer Science Ontology (CSO) [37] as an external resource. CSO is a standard ontology developed by The Open University's KMI lab. This ontology was developed from sixteen million scientific publications from the computer science domain and consists of fourteen thousand topics and more than one hundred sixty thousand semantic relationships.

5.3 Evaluation Measures

Many evaluation metrics have been developed to measure the performance of document indexing schemes. We employed three evaluation metrics, namely precision, recall and F-measure, in this work for comparison of the SNNOntoSDI approach with state-of-the-art models. Precision is the ratio of relevant concepts retrieved over total concepts retrieved, whereas recall is the ratio of relevant concepts retrieved over total relevant concepts. The harmonic mean of precision and recall is known as F-measure, which combines them into a single metric. Concepts assigned by a domain expert to each document are considered relevant. The mathematical expressions for these evaluation metrics are shown in Eqs. (7)–(9).

$$\text{Precision}(\text{Pr}) = \frac{\#\text{ relevant results retrieved}}{\#\text{ retrieved results}} \tag{7}$$

$$\text{Recall }(\text{Rc}) = \frac{\#\text{ relevant results retrieved}}{\#\text{ relevant results}} \tag{8}$$

$$F - measure = 2 * \frac{\text{Pr} * \text{Rc}}{\text{Pr} + \text{Rc}} \tag{9}$$

5.4 Parameter Tuning for Word Embedding Model

The word embedding model was trained on 4.65 million research publications in the computer science domain from Microsoft Academic Graph [36]. Then, Skip-gram with negative sampling approach [33,46] was applied to the training corpus to generate word embedding. For this, the authors preprocessed the training dataset and replaced spaces with underscores in all *n-grams* (bigrams and trigrams) matching the ontology concept (e.g., "rough set" became "rough_set" and "shallow neural network" became "shallow_neural_network"). The quality of the output of a ML model is heavily influenced by hyperparameter tuning. Tuning of hyperparameters plays a significant role in the output quality of a ML model. In our work, the hyperparameters for the SGNS model were tuned experimentally. The vector dimension is set to 128, whereas the window size and number of negative samples are set to 10 and 5 respectively. Similarly, the other hyperparameters *viz*. iteration value (for MAG corpus) and minimum

cut-off count are fixed at 5 and 10 respectively. Tab. 2 provides more details on hyperparameter tuning for word embedding model used in proposed work.

Table 2: Training parameters for Skip-gram model with negative sampling

Hyperparameter	Value
Vector dimension (embedding size)	128
Context window size	10
Negative sampling	5
Maximum iteration	5
Minimum count cut-off	10

5.5 Parameters Computation for Assigning Concept Weight

The parameters for the assignment of concept weights are calculated using the analytic hierarchy process (AHP). The AHP is a Multiple-Criteria Decision Analysis (MCDA) methodology [47], which is a relative measurement-based approach to quantitatively evaluate one alternative over others. The establishment of a hierarchical structure of weights is presented in Eq. (10) and the values of parameters in the matrix are defined by an expert based on the importance of different features used for concept weight calculation. In this work, the importance of semantic knowledge is $1/3$ of statistical knowledge, while scientific Named Entity knowledge is $1/5$ of statistical knowledge.

Here we calculate the values of parameters α, β and γ used in Eq. (3) for three features of the concept: statistical knowledge, semantic knowledge, and scientific Named Entity knowledge.

$$\begin{array}{c} \\ \alpha \\ \beta \\ \gamma \end{array} \begin{array}{c} \begin{array}{ccc} \alpha & \beta & \gamma \end{array} \\ \begin{pmatrix} 1 & 1/3 & 1/5 \\ 1/3 & 1 & 1/5 \\ 1/5 & 1/3 & 1 \end{pmatrix} \end{array} \tag{10}$$

The values of the parameters α, β and γ calculated using AHP are shown in Tab. 3.

Table 3: The values of parameters for concept weight calculation using the AHP

Parameter	α	β	γ
Weight	0.63699	0. 25828	0.10472

6 Results and Discussions

As discussed in Section 5, we tuned parameters for the proposed approach and presented the results with these parameters on five standard datasets. We compared the SNNOntoSDI approach with two state-of-the-art models: Text2Onto [48] and CFinder [49]. Both these tools are open source and state-of-the-art statistical methods with domain knowledge. Three variations of the proposed approach based on the implementation of the concept extraction module (syntactic, semantic and combined) were considered for the evaluation task. The first version of the proposed method, SNNOntoSDISyn, consists only of the syntactic module, and second version, SNNOntoSDISem, consists only of the semantic module, whereas the proposed method,

SNNOntoSDI, includes both the syntactic and semantic modules for concept extraction. In this section, the proposed approach is empirically analysed based on evaluation measures for the document indexing task.

The evaluation results for each method have been summarized in Tabs. 4–6. The bold values in the tables indicate the best value of the precision, recall and F-measure produced by the specific approach. Tabs. 4 and 5 illustrate the precision coefficient and recall values for compared methods on CACM, CISI, LISA, KDD and Inspec datasets. In comparison to Text2Onto and CFinder, the SNNOntoSDI approach produces higher values for precision on these benchmark datasets.

Table 4: Precision coefficient on CACM, CISI, LISA, KDD and Inspec datasets

Model	Dataset				
	CACM	CISI	LISA	KDD	Inspec
Text2Onto	0.441	0.404	0.412	0.377	0.436
CFinder	0.555	0.504	0.534	0.475	0.521
SNNOntoSDISyn	**0.782**	**0.775**	**0.803**	**0.756**	**0.763**
SNNOntoSDISem	0.691	0.702	0.712	0.657	0.677
SNNOntoSDI	0.763	0.754	0.752	0.707	0.732

Table 5: Recall values on CACM, CISI, LISA, KDD and Inspec datasets

Model	Dataset				
	CACM	CISI	LISA	KDD	Inspec
Text2Onto	0.474	0.485	0.465	0.491	0.507
CFinder	0.507	0.456	0.486	0.455	0.486
SNNOntoSDISyn	0.643	0.608	0.629	0.636	0.624
SNNOntoSDISem	**0.751**	**0.762**	**0.748**	**0.727**	**0.751**
SNNOntoSDI	0.722	0.697	0.715	0.705	0.705

Table 6: F-measure on CACM, CISI, LISA, KDD and Inspec datasets

Model	Dataset				
	CACM	CISI	LISA	KDD	Inspec
Text2Onto	0.456	0.440	0.437	0.426	0.469
CFinder	0.530	0.479	0.509	0.465	0.503
SNNOntoSDISyn	0.705	0.681	0.705	0.691	0.686
SNNOntoSDISem	0.719	**0.730**	0.729	0.690	0.712
SNNOntoSDI	**0.742**	0.724	**0.733**	**0.696**	**0.718**

The SNNOntoSDI approach improved the precision coefficient by 32.2%, 35%, 34%, 33% and 29.6% when compared to Text2Onto and CFinder on CACM, CISI, LISA, KDD and Inspec datasets respectively. The SNNOntoSDI approach outperformed the Text2Onto and CFinder on the recall metric by 24.8%, 21.2%, 25%, 21.4% and 19.8% on CACM, CISI, LISA, KDD and Inspec datasets respectively. Tab. 6 depicts the comparison of F-measure for compared methods on all five benchmark datasets. When compared to Text2Onto and CFinder, the SNNOntoSDI approach enhanced F-measure by 28.6%, 28.4%, 29.6%, 27% and 24.9% on CACM, CISI, LISA, KDD and Inspec datasets respectively.

Because of the syntactic match between document terms and ontology concepts, SNNOntoSDISyn has the highest precision value of the three variations of the proposed model. This method is quite good at extracting highly relevant concepts that are directly mentioned in the paper. The SNNOntoSDI demonstrates precision better than the SNNOntoSDISem, but lower than the SNNOntoSDISyn. As a result, the SNNOntoSDISem extracts semantically inferred concepts that are not explicitly addressed in the document, although it is prone to false positives. SNNOntoSDI demonstrated a slightly better recall value than the SNNOntoSDISyn but fell behind SNNOntoSDISem. Due to the inclusion of a high number of semantically identical concepts, SNNOntoSDISem performed worse on precision than SNNOntoSDISyn, but did better on recall and F-measure.

The results show that the proposed method performed better than Text2Onto and CFinder on the average precision metric on all five datasets. This performance is attributed to three features (statistical knowledge, semantic knowledge and scientific Named Entity) based on concept weight assignment in the proposed model, which assign top rankings to the relevant concepts, while Text2Onto and CFinder failed to assign rankings to the most relevant concepts when a concept appears multiple times in a document. In Text2Onto, concept weights were assigned using TF-IDF. If a domain relevant concept is present multiple times in a document, it may be assigned a lower weight based on its frequency. In CFinder, concept weights were calculated using a modified TF-IDF. The adjectives are removed from the concepts during the concept enrichment phase, leading to repetitive concepts. As a result, weight assignment to these concepts is not appropriate and leads to concept ranking issues. The capability of the proposed approach to learning term semantics based on the word2vec model and domain ontology produced higher recall values when compared to Text2Onto and CFinder.

7 Conclusions and Future Work

In this paper, we introduced a novel approach to semantic document indexing for Information Retrieval. For concept extraction, the proposed approach combined the Shallow Neural Network and domain ontology, which improved the degree of similarity between a document and ontology concepts. The Skip-gram model enhanced the capability of the proposed approach to recognize semantically linked concepts in document text. The morphological variants of concepts are included using a partial mapping between document terms and ontology concepts. The relevance of a concept to a document is modeled by assigning weights to the concept based on three features: statistical knowledge, semantic knowledge and scientific Named Entity knowledge. The parameters for these feature weights are calculated by employing Analytic Hierarchy Process. The empirical evaluation of the proposed approach on five benchmark datasets indicates that the proposed method produced high values of precision, recall and F-measure, indicating its better performance compared to two state-of-the-art methods, Text2Onto and CFinder. The proposed method used syntactic, semantic and scientific Named Entity features for assigning weight to concepts, which provided appropriate ranking to the relevant concepts.

The SNNOntoSDI approach can be applied in any other domain without constraint; only prerequisites are domain ontology and Skip-gram model trained on domain corpus. The hyperparameters of Skip-gram are corpus and problem-dependent. In future work, we will explore optimization algorithms for tuning

hyperparameters (negative sampling distribution, maximum iterations, sub-sampling parameters, window size and vector dimensions) of the word2vec model for performance enhancement. In addition, Knowledge Graph embedding may be explored for the concept extraction task. We have also planned to test our proposed indexing scheme with ontology and big data corpora from other scientific domains.

Funding Statement: The authors received no specific funding for this research work.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. F. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai *et al.,* "A deep look into neural ranking models for information retrieval," *Information Processing & Management*, vol. 57, no. 6, pp. 1–20, 2019.

[2] A. Sharma and S. Kumar, "Bayesian rough set based information retrieval," *Journal of Statistics and Management Systems*, vol. 23, no. 7, pp. 1147–1158, 2020.

[3] J. Jo, S. Lee, C. Lee, D. Lee and H. Lim, "Development of fashion product retrieval and recommendations model based on deep learning," *Electronics*, vol. 9, no. 3, pp. 508, 2020.

[4] N. Bhadwal, P. Agrawal and V. Madaan, "A machine translation system from Hindi to Sanskrit language using rule based approach," *Scalable Computing: Practice and Experience*, vol. 21, no. 3, pp. 543–553, 2020.

[5] P. Upadhyay, S. Bedathur, T. Chakraborty and M. Ramanath, "Aspect-based academic search using domain-specific KB," in *42nd European Conf. on Information Retrieval (ECIR)*, Lisbon, Portugal, vol. 12036, pp. 418–424, 2020.

[6] Y. Jiang, "Semantic search exploiting formal concept analysis, rough sets, and Wikipedia," *International Journal on Semantic Web and Information Systems*, vol. 14, no. 3, pp. 99–119, 2018.

[7] K. A. Fitzgerald, A. C. Harpe and C. S. Uys, "The hybridised indexing method for research-based information retrieval," *Journal of Information Science*, vol. 4, pp. 016555152199980, 2021. https://doi.org/10.1177/0165551521999800.

[8] K. Goel, C. Gupta, R. Rawal, P. Agrawal and V. Madaan, "FaD-CODS fake news detection on COVID-19 using description logics and semantic reasoning," *International Journal of Information Technology and Web Engineering*, vol. 16, no. 3, pp. 1–20, 2020.

[9] S. W. Fan-Jiang, T. H. Lo and B. Chen, "Spoken document retrieval leveraging BERT-based modeling and query reformulation," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Spain, pp. 8144–8148, 2020.

[10] N. Limbasiya and P. Agrawal, "Bidirectional long short-term memory-based spatio-temporal in community question answering," in *Deep Learning-Based Approaches for Sentiment Analysis*, Singapore, Springer Nature, pp. 291–310, 2002 https://link.springer.com/book/10.1007/978-981-15-1216-2

[11] H. Kammoun, I. Gabsi and I. Amous, "MeSH-based semantic indexing approach to enhance biomedical information retrieval," *The Computer Journal*, vol. 10, pp. 115, 2020. https://doi.org/10.1093/comjnl/bxaa073.

[12] P. K. Verma, P. Agrawal, I. Amorim and R. Prodan, "WELFake: Word embedding over linguistic features for fake news detection," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 4, pp. 881–893, 2021.

[13] T. Prasanth and M. Gunasekaran, "Effective big data retrieval using deep learning modified neural networks," *Mobile Networks and Applications*, vol. 24, no. 1, pp. 282–294, 2019.

[14] H. Arezki and M. Boughanem, "Term position based language model for information retrieval," *Journal of the Association for Information Science and Technology*, vol. 7, no. 9, pp. 1–16, 2020.

[15] A. Esteva, A. Kale, R. Paulus, K. Hasimoto, W. Yin *et al.,* "COVID-19 information retrieval with deep-learning based semantic search, question answering, and abstractive summarization," *npj Digital Medicine*, vol. 4, no. 68, pp. 102067, 2021.

[16] N. Ebadi, M. Jozani, K. K. R. Choo and P. Rad, "A memory network information retrieval model for identification of news misinformation," *IEEE Transactions on Big Data*, pp. 1, 2021. https://doi.org/10.1109/TBDATA.2020.3048961.

[17] P. Mahalakshmi and N. S. Fatima, "Ensembling of text and images using deep convolutional neural networks for intelligent information retrieval," *Wireless Personal Communications*, vol. 24, no. 4, pp. 694, 2021. https://doi.org/10.1007/s11277-021-08211-x.

[18] A. Gupta and D. Yadav, "A novel approach to perform context-based automatic spoken document retrieval of political speeches based on wavelet tree indexing," *Multimedia Tools & Applications*, vol. 80, no. 14, pp. 22209–22229, 2021.

[19] S. Wagenpfeil, F. Engel, P. M. Kevitt and M. Hemmje, "AI-based semantic multimedia indexing and retrieval for social media on smartphones," *Information-an International Interdisciplinary Journal*, vol. 12, no. 43, pp. 1–30, 2020.

[20] Y. Djenouri, A. Belhadi, D. Djenouri and J. C. Lin, "Cluster-based information retrieval using pattern mining," *Applied Intelligence*, vol. 51, no. 4, pp. 1888–1903, 2021.

[21] A. P. Bhopale and A. Tiwari, "Swarm optimized cluster based framework for information retrieval," *Expert Systems with Applications*, vol. 154, no. 2, pp. 113441, 2020.

[22] S. Sharma, V. Gupta and M. Juneja, "Diverse feature set based keyphrase extraction and indexing techniques," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4111–4142, 2021.

[23] D. Qiu, H. Jiang and S. Chen, "Fuzzy information retrieval based on continuous bag-of-words model," *Symmetry*, vol. 12, no. 2, pp. 225, 2020.

[24] Z. Dai and J. Callan, "Deeper text understanding for IR with contextual neural language modelling," in *42nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'19)*, France, pp. 985–988, 2019.

[25] S. Zouaoui and K. Rezeg, "A novel Quranic search engine using an ontology-based semantic indexing," *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 3653–3674, 2021.

[26] M. Subramaniam, A. Kathirvel and H. A. Basha, "Modified firefly algorithm and fuzzy c-mean clustering based semantic information retrieval," *Journal of Web Engineering*, vol. 20, no. 1, pp. 33–52, 2020.

[27] K. Boukhari and M. N. Omri, "DL-VSM based document indexing approach for information retrieval," *Journal of Ambient Intelligence and Humanized Computing*, vol. 16, no. 1, pp. 138, 2020. https://doi.org/10.1007/s12652-020-01684-x.

[28] R. Rahimi, A. Montazeralghaem and A. Shakery, "An axiomatic approach to corpus-based cross-language information retrieval," *Information Retrieval Journal*, vol. 23, no. 3, pp. 191–215, 2020.

[29] Y. Jiang, "Semantically-enhanced information retrieval using multiple knowledge sources," *Cluster Computing*, vol. 23, no. 4, pp. 2925–2944, 2020.

[30] M. Tang, J. Chen, H. Chen, Z. Xu, Y. Wang *et al.,* "An ontology-improved vector space model for semantic retrieval," *The Electronic Library*, vol. 38, no. 5/6, pp. 919–942, 2020.

[31] B. Yu, "Research on information retrieval model based on ontology," *EURASIP Journal on Wireless Communications and Networking*, vol. 30, no. 1, pp. 1–8, 2019.

[32] A. Sharma and S. Kumar, "Semantic web-based information retrieval models: A systematic survey," in *Int. Conf. on Recent Developments in Science, Engineering and Technology (REDSET)*, Gurugram, India, vol. 1230, pp. 204–222, 2019.

[33] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," in *1st Int. Conf. on Learning Representations (ICLR)*, Arizona, USA, 2013 arXiv preprint arXiv:1301.3781

[34] E. Loper and S. Bird, "NLTK: The natural language toolkit," in *Association for Computational Linguistics Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics (ETMTNLP)*. Pennsylvania, USA, Vol. 1. 63–70, 2002.

[35] X. Jiang and A. H. Tan, "CRCTOL: A semantic-based domain ontology learning system," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 1, pp. 150–168, 2010.

[36] A. A. Salatino, F. Osborne, T. Thanapalasingam and E. Motta, "The CSO classifier: ontology-driven detection of research topics in scholarly articles," in *23rd Int. Conf. on Theory and Practice of Digital Libraries (TPDL)*, Oslo, Norway, vol. 11799, pp. 296–311, 2019.

[37] A. A. Salatino, F. Osborne, T. Thanapalasingam and E. Motta, "The computer science ontology: A large-scale taxonomy of research areas," in *17th Int. Semantic Web Conf. (ISWA)*, Monterey, CA, USA, pp. 187–205, 2018.

[38] L. Havrlant and A. Kreinovich, "A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation)," *International Journal of General Systems*, vol. 16, no. 1, pp. 27–36, 2017.

[39] A. Goyal, V. Gupta and M. Kumar, "Recent named entity recognition and classification techniques: A systematic review," *Computer Science Review*, vol. 29, no. 1, pp. 21–43, 2018.

[40] Y. Luan, L. He, M. Ostendorf and H. Hajishirzi, "Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction," in *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium, pp. 3219–3232, 2018.

[41] Information Retrieval Group, "University of Glasgow, UK, Information Retrieval Test Collections, CACM dataset.," 2022. [Online]. Available: http://ir.dcs.gla.ac.uk/resources/test_collections/cacm/.

[42] Information Retrieval Group, "University of Glasgow, UK, Information Retrieval Test Collections, CISI dataset," 2022. [Online]. Available: http://ir.dcs.gla.ac.uk/resources/test_collections/cisi/.

[43] Information Retrieval Group, "University of Glasgow, UK, Information Retrieval Test Collections, LISA dataset," 2022. [Online]. Available: http://ir.dcs.gla.ac.uk/resources/test_collections/lisa/.

[44] C. Caragea, F. A. Bulgarov, A. Godea and S. D. Gollapalli, "Citation-enhanced keyword extraction from research papers: a supervised approach," in *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1435–1446, 2014.

[45] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Pennsylvania, USA, pp. 216–223, 2003.

[46] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," in *26th Int. Conf. on Neural Information Processing Systems (NIPS)*, Nevada, USA, vol. 2, pp. 3111–3119, 2013.

[47] T. Saaty, "A scaling method for priorities in hierarchical structures," *Journal of Mathematical Psychology*, vol. 15, no. 3, pp. 234–281, 1977.

[48] P. Cimiano and J. Völker, "Text2Onto: A framework for ontology learning and data-driven change discovery," in *10th Int. Conf. on Applications of Natural Language to Information Systems (NLDB)*, Alicante, Spain, pp. 227–238, 2005.

[49] Y. B. Kang, P. D. Haghighi and F. Burstein, "CFinder: An intelligent key concept finder from text for ontology development," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4494–4504, 2014.