Post-print article version published in Information Processing & Management, Vol 56, Issue 5, 2019.
DOI: https://doi.org/10.1016/j.ipm.2019.05.003

The impact of deep learning on document classification using semantically rich representations

Zenun Kastrati*, Ali Shariq Imran, Sule Yildirim Yayilgan

NORWEGIAN University of Science AND Technology, NORWAY

ARTICLEINFO

Keywords:
Document representation
Document classification
Deep learning
Ontology
Machine learning

ABSTRACT

This paper presents a semantically rich document representation model for automatically classifying financial documents into predefined categories utilizing deep learning. The model architecture consists of two main modules including document representation and document classification. In the first module, a document is enriched with semantics using background knowledge provided by an ontology and through the acquisition of its relevant terminology. Acquisition of terminology integrated to the ontology extends the capabilities of semantically rich document representations with an in depth-coverage of concepts, thereby capturing the whole conceptualization involved in documents. Semantically rich representations obtained from the first module will serve as input to the document classification module which aims at finding the most appropriate category for that document through deep learning. Three different deep learning networks each belonging to a different category of machine learning techniques for ontological document classification using a real-life ontology are used.

Multiple simulations are carried out with various deep neural networks configurations, and our findings reveal that a three hidden layer feedforward network with 1024 neurons obtain the highest document classification performance on the INFUSE dataset. The performance in terms of F1 score is further increased by almost five percentage points to 78.10% for the same network configuration when the relevant terminology integrated to the ontology is applied to enrich document representation. Furthermore, we conducted a comparative performance evaluation using various state-of-the-art document representation approaches and classification techniques including shallow and conventional machine learning classifiers.

1. Introduction

The early 2000s have seen extensive use of ontological representations for actually being able to represent and model the relevant knowledge in a specific domain and make it accessible across various applications of the domain and for improving document classification performance, particularly on the web. Ontological representation techniques provide semantic understanding of documents by using an ontology to identify and locate concepts in these documents. On the other hand, deep learning has been a major endeavor in various computer science fields mainly for enhancing the learning performance, with a special focus on the classification.

Even though these two aspects, semantic enrichment of document representation through ontologies and classification using deep learning, have been addressed separately by many research works and have been shown to be useful for classification in general

* Corresponding author.
E-MAIL ADDRESS: zenun.kastrati@lnu.se (Z. Kastrati).

https://doi.org/10.1016/j.ipm.2019.05.003

(Bing, Jiang, Lam, Zhang, & Jameel, 2015; Kim, Kim, Kim, & Lim, 2018; Kowsari et al., 2017; Sanchez-Pi, Marti, & Garcia, 2014; 2016; Zhang, Du, Yoshida, & Wang, 2018). However, an investigation into the integration (ensemble) of these two aspects for document classification in particular is either lacking or insufficiently addressed in the literature. Particularly, document representation which is one of the crucial factors that determines the performance of ontology-based classification models has not been established well. Consequently, documents are represented as vectors containing relevance of the concepts that are gathered by an ontology by searching only the presence of their lexicalizations (concept labels) in the documents. This limits the capabilities of classification models to capture the whole conceptualization involved in documents. Therefore, this paper addresses this issue by proposing a classification model in which a document representation capable of capturing the entire semantic information contained in documents is integrated with deep learning for classifying documents. Basically, the proposed document classification model is composed of two main modules.

The first module consists of semantic enrichment of document representation using background knowledge derived from an ontology, and through the relevant terminology integrated into the ontology. We partially investigated into this aspect in our previous research (Kastrati & Yayilgan, 2017). Background knowledge provided by an ontology is embedded into a document using a matching technique. The basic idea of this technique is simply mapping terms to concepts by searching only for concepts in the ontology that have labels matching either fully or partially with a term in the document. In addition, the relevant terminology attached to the ontology is located and acquired into a document by combining its contextual and semantic information. In our previous work, we have shown that representing documents using the terminology integrated to an ontology improves document classification performance (Kastrati & Yayilgan, 2017).

The second module of the proposed classification model contains the deep learning classifier which is provided with semantically rich document representations built in the first module to classifying the documents into a predefined class label accordingly. Three deep learning techniques, namely feed-forward Multilayer perceptron, Long-short term memory and Convolutional neural network (Goodfellow, Bengio, & Courville, 2016), are employed which are introduced and discussed in Section 5.2.

Several simulations on a real-life INFUSE dataset along with its baseline domain ontology are carried out to demonstrate the applicability of our proposed approach utilizing semantically rich representations and deep learning based classification model and to validate its accuracy. Extensive experimental results demonstrate a significant improvement of the classification performance when using semantically rich document representation and show that deep learning techniques outperform the conventional machine learning techniques, achieving better performance in every point of testing.

The remaining of the paper is organized as the following. Section 2 provides purpose and objectives of this study. In Section 3, we provide the related work about document classification analyzed in the perspective of document representation enrichment and classification using deep learning. Section 4 encompasses the deep learning architecture that is proposed for document classification. Section 5, provides our experiments followed by results and analysis given in Section 6. We conclude the paper with Section 7 while giving our conclusions and proposals for future work.

2. Purpose and objectives

The main purpose of this research work is to develop a semantic based document classification model that exploits ontologies for semantically rich document representations and takes advantage of deep learning to improve classification performance. In this context, we specifically formulated the following three objectives:

Enrich document representation with semantics using an ontology

The core of this objective is to describe the essential steps of enriching document representation with semantics using background knowledge provided by an ontology and through extraction of the lexical information, i.e., synonyms, linguistic variants, etc., that can be integrated to that ontology.

Evaluate classification performance using various deep learning networks and different levels of document representation

The primary focus of this objective is to perform an in-depth investigation and analysis of performance measurement of document classification using proposed semantically rich document representations. Three different deep learning architecture configurations, namely Feedforward network, Recurrent neural network, and Convolutional network, will be employed and tested on various levels of document representation. Moreover, a performance comparison of our proposed document representation technique with three state-of-the-art representation techniques including *tf*idf*, word embedding, and topic modeling, will be conducted.

Compare conventional machine learning and deep learning techniques

This objective focuses on conducting a comprehensive comparative evaluation of the performance of deep learning networks for document classification with that of shallow and more conventional machine learning techniques including Support vector machine, Naive Bayes, and Decision tree. In particular, the focus is to compare and contrast the performance of these techniques for a number of document representations.

3. Related work

Over the past few years, an increasing interest has been shown in the study of semantic-based document classification, with a particular focus placed on semantic enrichment of document representation using background knowledge exploited by ontologies and taxonomies. In essence, all these studies use semantic concepts either extracted from ontologies or taxonomies to map a document

from a keyword vector to a concept vector for capturing the semantic information contained in the document. For example, Wu et al. (2017) proposed an efficient approach to text document classification which relies on Wikipedia taxonomy for document representation. In this classification approach, each document is mapped to a concept vector composed of a set of semantic concepts gathered by Wikipedia reference space through a Wikipedia matching technique (Pak & Chung, 2010). The set of relevant concepts that are extracted using several heuristic selection rules avoid the need to conduct time-consuming full document matching over all the Wikipedia concepts and hence improving the efficiency of concept vectors generation. Cagliero and Garza (2013) also proposed a classification approach that relies on taxonomy information for enriching data representation with semantics. In particular, they developed a general-purpose strategy to improve classification accuracy by supplementing textual data with semantics using background knowledge, i.e., *IS-A* relationships, provided by a taxonomy.

There is some other research work in which background knowledge exploited by ontologies is extensively used for enriching documents representation with semantics. For instance, the work by Sanchez-Pi et al. (2014) introduced a classification approach for classifying accidents from the oil and gas industry using background knowledge provided by means of an ontology. Notably, the background knowledge provided by the domain ontology for Health, Safety, and Environment for oil and gas application contexts is expanded with a thesaurus for finding non-explicit relations to make the approach more flexible and resilient to classifying real-life documents which can be written in a heterogeneous form. Later, an extension of this classification approach is introduced by Sanchez-Pi, Marti, and Garcia (2016), in which a list of technical terms generated in a semi-automatic way using an n-gram extraction technique is used in addition to the background knowledge derived by the ontology. The study conducted by Bing et al. (2015) proposed an Adaptive Concept Resolution for document representation. The model uses a set of concepts gathered from different levels of the structure of an ontology using the border. The border is a cross section in the ontology structure, and it indicates the depth of concepts to be considered, i.e., concepts below the border are merged into one of the concepts on the border. The border is a tailor-made semantic concept representation for a document, and it is defined using information gain.

Utilizing ontologies as a means for enriching document representation with semantics from the biomedical domain is also acknowledged by Camous, Blott, and Smeaton (2007), Dinh and Tamine (2011) and Sy et al. (2012). Camous et al. (2007) presented an ontology-based classification approach in which Medical Subject Headings (MeSH) ontology is employed to enrich the existing MeSH-based representation of documents with semantics. New terminology which is semantically related to the initial document representation is located and extracted from the document using a semantic similarity measure based on the MeSh hierarchy. Dinh and Tamine (2011) presented a similar document classification approach that also relies on the MeSH ontology for semantic representation of documents, but it employs a content-based cosine similarity measure to acquire domain concepts. Background knowledge provided by the MeSh ontology for enriching biomedical document representation is also used by an ontology-based system presented by Sy et al. (2012).

Observing the recent text classification literature, we can see that a growing body of research has examined deep learning techniques for document classification. For example, an ontology-based deep learning model for prediction of human behavior is proposed by Phan, Dou, Wang, Kil, and Piniewski (2017). The model relies on health ontologies to learn user representation from health social networks, and it aims to replicate the original structure of personal characteristics. In addition to user representation, the model feeds deep learning with human behavior determinants such as self-motivation, social influences, and environmental events to improve human behavior prediction accuracy. Their experimental results demonstrate that their classification model achieves higher effectiveness compared with conventional methods. Although the idea presented in this study is similar to our work, we focus on representing documents using broader coverage of concepts including the acquisition of new terminology. Exploring social media to detect disease outbreaks using deep learning is also acknowledged by Serban, Thapen, Maginnis, Hankin, and Foot (2018). Specifically, the authors developed a system that applies deep learning to classify health-related tweets accurately. The system initially detects illness outbreaks by exploring Twitter data and then provides to health officials the relevant information about these outbreaks. The study conducted by Hassan and Mahmood (2017) did sentiment analysis using a Long short-term memory (LSTM) and pre-trained word vectors validating their approach on two benchmark datasets. Their results show that LSTM with one single layer gives the best performance when used with unsupervised word vectors. Hassan and Mahmood (2018) use convolutional recurrent deep learning model for sentence classification. The research conducted by Agarwal, Ramampiaro, Langseth, and Ruocco (2018) acknowledges semantic representation of sentences. They developed a model called DeepParaphrase which relies on CNN and RNN to create an informative semantic representation of each sentence. Specifically, CNN is used to extract the local region information, i.e., relevant n-grams from the sentence while RNN is used to capture the long-term dependency information.

In a study conducted by Kim (2014), feature vectors formed by words in a sentence are provided as input to convolutional nets, and several CNN models are tested against other classification methods to clarify the CNN variation that is most promising. Variations of CNN outperform other models in 4 of the datasets out of 6 of them. HDLTex is developed and tested on three datasets (Kowsari et al. (2017) where deep hierarchical architectures are proposed. HDLTex is shown to outperform SVM and non-hierarchical deep methods. Wei et al. (2018) proposed a deep learning technique called RNN-LSTM for processing malfunction inspection report. This classification technique employs RNN with LSTM and its training strategy involves two phases. The first phase replicates targets at each sequence step, and in the second phase, the corresponding fault class labels are predicted. The predicted labels are compared with the original data labels to compute the classification accuracy. Zhang et al. (2018) investigate deceptive opinions on the internet using word contexts and deep learning. They proposed a binary classification model called Deceptive Review Identification by Recurrent Convolutional Neural Network (DRI-RCNN) which differentiate the deceptive and truthful contextual knowledge inserted in the online reviews. The model represents each word in a review with six components as a recurrent convolutional vector. The first and second components are word vectors derived from training deceptive and truthful reviews. The left neighboring deceptive and truthful context vectors constitute the third and fourth components of the vector while the fifth and six components represent right

neighboring deceptive and truthful context vectors.

In contrary to the performance of deep learning approaches reported in the research presented above, Kim et al. (2018) proposed a novel text classification approach relying on semantic Naive Bayes with tensor space model for document representation which outperforms new deep learning based classification approaches. The method employs Wikipedia encyclopedia as an external knowledge source to enrich document representation semantically. Specifically, a semantic concept is defined through a concept-level informative Wikipedia page. The classification approach is tested on three different benchmark datasets, and it has shown higher performance than three deep learning methods, namely DNN, CNN, and RNN. Similar findings have been reported by Yang et al. (2016) who found that the CNN models tested do not particularly provide outstanding performance in comparison to Hierarchical Attention Networks for Document Classification. The CNN models are either from Kim (2014) or from Zhang, Zhao, and LeCun (2015) where Character level CNN models called CNN-char are reported. A detailed description of the past and recent advancements in semantic document classification which relies on knowledge-based sources, i.e., ontology or taxonomy and deep learning to enrich document representation are explored in the survey conducted by Altinel and Ganiz (2018).

In contrast to the aforementioned approaches, our proposed classification approach takes advantage of the strengths of both semantically rich document representations and deep learning techniques and use them to improve classification effectiveness in terms of accuracy. From the document representation perspective, our approach exploits ontologies and the terminology that can be integrated to these ontologies which enable to shift from literal (keyword) based document representation toward semantic (concept) based document representation. Consequently, acquisition of relevant terminology that can be attached to an ontology provides broader coverage of document representation rather than using only the ontology concepts. Additionally, a real-life ontology that comes from the financial domain is used for enriching with semantics document representation. From the deep learning perspective, our proposed approach employs various deep learning architecture configurations and test them on different document representation models. Besides, it also employs conventional machine learning techniques and compares their performances.

4. System overview

The architecture of our model is composed of two main modules: 1) document representation with a special focus on semantic representation of documents, and 2) document classification. The model architecture is illustrated in Fig. 1.

4.1. Document REPRESENTATION

The input of the first module of the proposed classification model is a collection of documents stored in unstructured textual

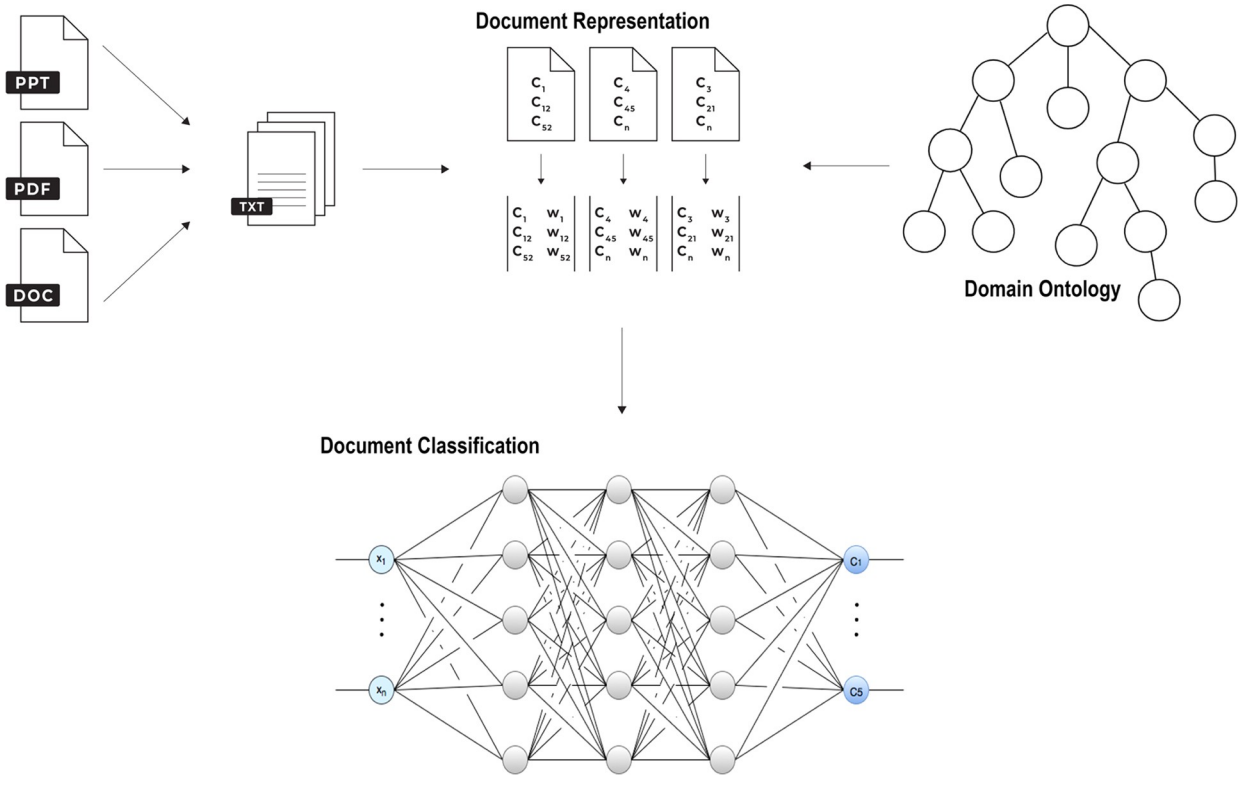

Fig. 1. Architecture of the proposed ontology-based document classification model.

formats such as Word, PDF, Powerpoint. These documents are then converted into plain texts as shown in Fig. 1. At this stage, the documents do not contain any semantics associated with them.

Representing documents as a feature vector using a vector space representation model (Keikha, Khonsari, & Oroumchian, 2009) is the next step of our document classification model. Feature vectors are constructed using background knowledge gathered by a domain ontology so as to make a step away from the keyword-based representation towards the semantic-based representation. In semantic-based document representation, each document is enriched with semantics embedded using:

1. the matching technique in which terms appearing in a document are mapped with the relevant concepts from the domain ontology, and
2. identification and acquisition of relevant terminology that can be integrated/attached to concepts of the domain ontology.

Embedding semantics into a document using matching technique is a simple and straightforward process. It basically tries to associate terms extracted from documents with concepts of the ontology. Terms are located and extracted from documents using an inverted indexing technique which generates a list of all unique terms that occur in any document and a set of documents in which these terms occur. The extracted terms are normalized using a stemming method. Next, noisy terms including terms with single character, stop words and punctuation are removed from the list of extracted terms. The extracted terms are associated with the concepts of the ontology by searching only for concepts that have labels matching either exactly or partially with a term occurring in the document. An exact match is a type of matching technique in which a concept label is identical with the term occurring in the document while in a partial match, a concept label contains terms occurring in the document. To measure the similarity between concept labels and extracted terms for both types of matching technique, exact and partial, a Levenshtein distance algorithm (Miller, Vandome, & McBrewster, 2009) is used.

The exact and partial match is formally defined as the following.

Definition 1. Let Ont be the domain ontology and let Doc be a document defined by a finite set of terms. Mapping of term $t_i \in Doc$ into concept $c_j \in Ont$ is defined as:

$$EM(t_i, c_j) = \begin{cases} \mathbf{1}, & if\ label(c_j) = t_i \\ \mathbf{0}, & if\ \ label(c_j) \neq t_i \end{cases}$$

$$PM(t_i, c_j) = \begin{cases} \mathbf{1}, & if\ label(c_j)\ contains\ t_i \\ \mathbf{0}, & if\ label(c_j)\ does\ not\ contain\ t_i \end{cases}$$

where, EM and PM denote exact match and partial match, respectively.

If $EM(t_i, c_j) = \mathbf{1}$, means that term t_i and concept label c_j are identical, and thus term t_i is replaced with concept c_j. For instance, for an ontology concept such as PARTICIPANT or CALL as shown in Fig. 3, there exists an identical term extracted from the document.

If $PM(t_i, c_j) = \mathbf{1}$, means that term t_i is part of concept label c_j, and thus term t_i is replaced with concept c_j. For instance, the *project_funding* compound concept shown in Fig. 3, contains terms extracted from the document such as *project* and/or *funding*.

Associating semantics with a document through identification and acquisition of terminology that is related and can be attached to ontology concepts is a more complex task that relies on exploiting both contextual and semantic information of terms occurring in a document.

Contextual information (Eq. (1)) of a term is defined by its surroundings, that is, the part of a text in which that particular term occurs and it is computed using cosine similarity between the feature vectors.

$$Context(t_i, t_j) = \frac{t_i * t_j}{\| t_i \| \| t_j \|} \quad (1)$$

The feature vectors t_i and t_j are composed of values derived by three statistical features, namely, frequency of occurrences of the term in corresponding document, its font types, and font sizes, respectively. Different font types, i.e. *bold,* ITALIC, *underline*, and font sizes, i.e. *title, level 1, level 2*, are introduced to derive the context. A linear model is adopted to set different values for various font types and font sizes in order to keep the effect of each feature the same for all values of the other features, e.g., *title* font size of terms reflects the same effect for every value of *level 1* or *level 2* font size of terms.

Semantic information of a term is defined by using a semantic similarity measure based on the English lexical database WordNet. Wu&Palmer similarity measure (Wu & Palmer, 1994) is employed to compute a semantic score (Eq. (2)) for all possible pairs of terms t_i and t_j occurring in a document. Terms t_i and t_j may have multiple senses (meanings), therefore a word sense disambiguation technique called predominant sense heuristic is employed to find the correct meaning of these terms. Predominant sense heuristic uses distributional property of senses assuming that correct meaning of a term is represented by the most common sense of that term.

$$Semantic(t_i, t_j) = \frac{\mathbf{2} * depth(lcs)}{depth(t_i) + depth(t_j)} \quad (2)$$

Parameter $depth(lcs)$ shows the least common subsumer of terms t_i and t_j, and parameters $depth(t_i)$ and $depth(t_j)$ show the path's depth of terms t_i and t_j in the WordNet.

Combining contextual and semantic information gives an aggregated score as shown in the following Equation:

$$AggregatedScore(t_i, t_j) = \lambda * Context(t_i, t_j) + (1 - \lambda) * Semantic(t_i, t_j) \quad (3)$$

Weighted parameter λ shows the contribution of each of the components i.e., context and semantic, on the aggregated score. In this paper, λ is set to 0.5 based on the empirical analysis conducted by Kastrati, Imran, and Yayilgan (2016).

Once the aggregated score is computed for all terms, a rank cut-off method is applied using a threshold to acquire terms that are related and can be integrated to concepts of the ontology. Terms that are above the specified threshold (top-N) are considered to be the relevant terms.

Next, a numeric value is assigned to each concept in order to show the discriminative power of concepts for distinguishing a document from the other documents. The numeric value is computed using concept weighting scheme proposed by Kastrati, Imran, and Yayilgan (2015). It is composed of two main factors, concept importance and concept relevance. These two factors reflect the discriminative power of concepts with respect to documents using frequency of occurrences and the position of concepts in the hierarchy structure of the ontology.

The output of document representation module will serve as input to the classification module which consists of deep learning.

4.2. Document CLASSIFICATION

The second module of the proposed model consists of document classification. Documents enriched with semantics are classified on a variety of deep neural networks (DNN) with different network architectures and configurations.

A DNN is a network that consists of at least three hidden layers with multiple nodes. By definition a DNN is a much wider and a deeper network containing successive layers of nodes. Often a multilayer perceptron (MLP) containing more than one hidden layer is considered as a baseline DNN. The training of a network is carried out in two phases including pre-training and fine tuning.

In the pre-training phase the weights of the network are initialized in an unsupervised manner. This is an important step that will affect how the network weights will converge during the training phase. The initial weights are estimated using a generative deep belief networks (DBN) on the input data (Hinton, Osindero, & Teh, 2006). The model is then trained in a greedy way by taking two layers at a time as a restricted Boltzmann machine (RBM) given as:

$$E(v, h) = -\sum_{k=1}^{K}\sum_{l=1}^{L}\frac{v_k}{\sigma_k}h_l w_{kl} - \sum_{k=1}^{K}\frac{(v_k - a_k)^2}{2\sigma_k^2} - \sum_{l=1}^{L} h_l b_l \quad (4)$$

where Eq. (4) is the energy function for the Gaussian-Bernoulli RMB, σ_k is the standard deviation, w_{kl} is the weight value connecting visible units v_k and the hidden units h_l, A_k and b_l are the bias for visible and hidden units respectively.

The joint probability of hidden and visible units is then defined as:

$$p(v, h) = \frac{e^{-E(v,h)}}{\sum_{v,k} e^{-E(v,h)}} \quad (5)$$

The trainable parameters are then estimated by maximizing the expected log probability using the contrastive divergence algorithm (Hinton et al., 2006), give as:

$$\theta = argmax_{\theta} E\left[log \sum_{h} p(v, h)\right] \quad (6)$$

where θ represents the weights, biases and standard deviation.

In the second phase the network parameters are adjusted in a supervised manner using a backpropagation technique. The labels are introduced in this case. A cross-entropy cost function is applied to update the randomly initialized weights at the output layer by maximizing the cross entropy between the labels and the estimated outputs.

5. Experiments

In this section, a description of the dataset, the domain ontology, and the architecture of deep networks used to conduct the experiments for demonstrating the applicability of our proposed model and to validate its efficacy in terms of classification performance are presented.

5.1. DATASET AND DOMAIN ontology

For the evaluation, we used a real-life dataset consisting of 467 grant documents that have been assembled and classified into 5 different categories by the field experts as part of the INFUSE[1] project. All documents are written in English language and stored in pdf format. The average length of a document is 13,146 words (tokens). The dataset is divided randomly in three parts: training, testing, and validation. Specifically, out of 467 documents of the dataset, 228 (50%) of the documents are used to train the classifier,

[1] https://www.eurostars-eureka.eu/project/id/7141

141 (30%) for testing, and the remaining 98 (20%) documents are used to validate the performance of the classifier. The number of documents varies widely from category to category, e.g., Society category contains 165 documents while Music category consists of only 14 documents. Fig. 2 illustrates the 5 categories along with the distribution of training, testing, and validation documents and the distribution of words per each category. Dataset[2] composed of feature vectors constructed using baseline ontology and its acquired relevant terminology is made open and available to the public.

The ontology used for experimenting in this paper is a real-life ontology that also comes from the funding domain. The ontology was developed as part of the INFUSE project. It consists of 85 concepts and 18 ontological relationships that connect these concepts. Ontological relationships are constituted by taxonomic, i.e., *IS-A*, and non-taxonomic relations, i.e., *APPLIESFOR, isReceivedBy*, etc. A part of the INFUSE domain ontology is illustrated in Fig. 3.

5.2. DNN Architecture

Three variants of DNN architecture are explored in this study. These are briefly explained in following subsections.

5.2.1. MULTILAYER perceptron

Multilayer perceptron (MLP) is a feed-forward network. All the neurons in one layer are fully connected to the neurons in the adjacent layer. The model uses a supervised learning technique called “backpropagation” to update the weights for training. A single hidden layer MLP is basically a vanilla network, so for an MLP to be a truly deep network it should have at least more than one hidden layer.

5.2.2. Long-short term memory

Long-short term memory (LSTM) is a DNN that belongs to a recurrent neural networks (RNN) category. This means that it has both forward and backward network connections. This model can also memorize the values of the previous layers. It uses previously learned information in calculating the weights and bias for the new layers which helps it to perform better for time dependent data samples.

5.2.3. CONVOLUTIONAL NEURAL network

Convolutional neural network (CNN) differs from the other two models in the sense that each layer in the CNN is a convolution operation, thus, the name ‘convolutional’ neural network. The weights in a CNN are shared between neurons which mimics the connectivity patterns between neurons of the animal visual cortex, a network truly inspired by the biological process. Normally, in a CNN, results from a single convolutional layer or multiple convolutional layers applied in succession are downsampled using a pooling layer to speed up the process. So, a max-pooling layer or a global pooling layer is often added after convolutional layers.

5.3. Model CONFIGURATION

Different configurations of DNN architectures are evaluated in this study by varying the number of layers and number of neurons in each layer to see which configuration gives best performance on the INFUSE dataset. A total of 15 different combinations are evaluated for MLP in which 3-, 5-, and 7-hidden layer networks with 64, 128, 256, 512, 1024 neurons are experimented with. The number of neurons is kept same in each layer for a single network configuration. For instance, if it is a 3-hidden layer network with 64 neurons, then each of the 3 hidden layers will have 64 neurons.

The input to the network is a feature vector extracted from the dataset. The size of the feature vector varies based on the level of document representation while the output of the network is a predicted class sample belonging to one of the five categories the document belongs to.

Fig. 4 shows the model architecture of a N-hidden layer fully-connected DNN with 64 neurons in each of the hidden layers. For the experimentation, the following parameters of the DNN are used: loss function: categorical cross entropy, learning rate: 0.01, optimizer: adam, activation function: rectified linear units (ReLU), batch size for training and prediction: 1024. A *SOFTMAX* function is used at the output layer. The number of trainable parameters for a 5-hidden layer DNN with an output size of 64 neurons is shown in Table 1.

6. Results and analysis

This section gives the results obtained using various deep learning and conventional machine learning techniques, and a comparison of these techniques. It also covers a comparative performance analysis of different document representation techniques.

6.1. Results of DNN on the INFUSE DATASET

The results for various configurations of the MLP architecture as described in Section 5.2 to determine which MLP architecture gives best performance on the INFUSE dataset, are presented in this subsection. Initial random seed is set to 1 for reproducibility of

[2] https://github.com/zenunk/Infuse

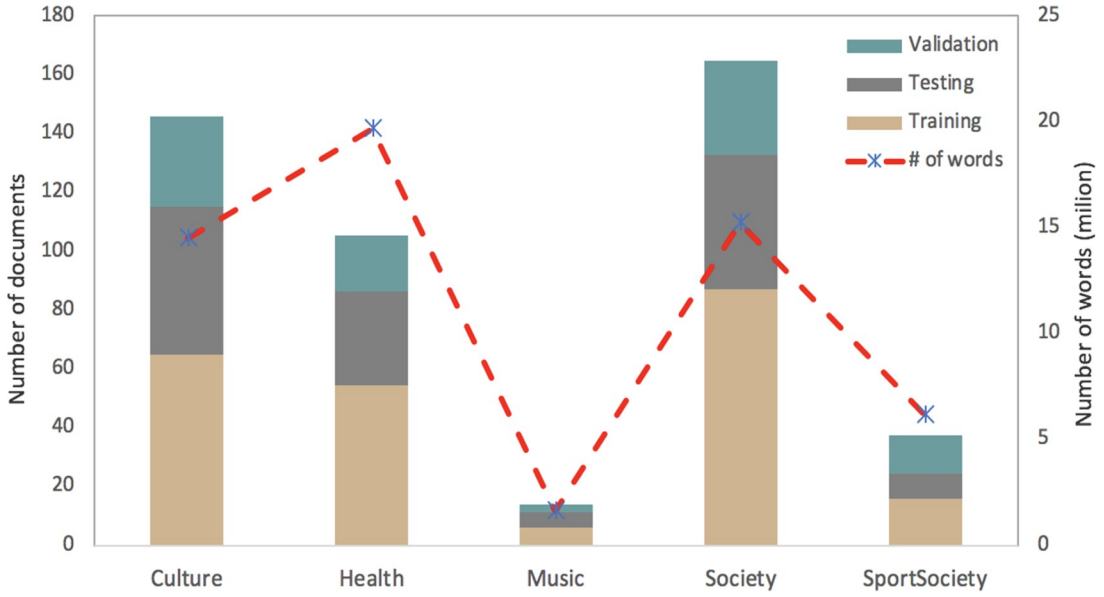

Fig. 2. Dataset illustrating the categories and the distribution of training, testing, and validation documents along with the number of documents and the number of words for each category.

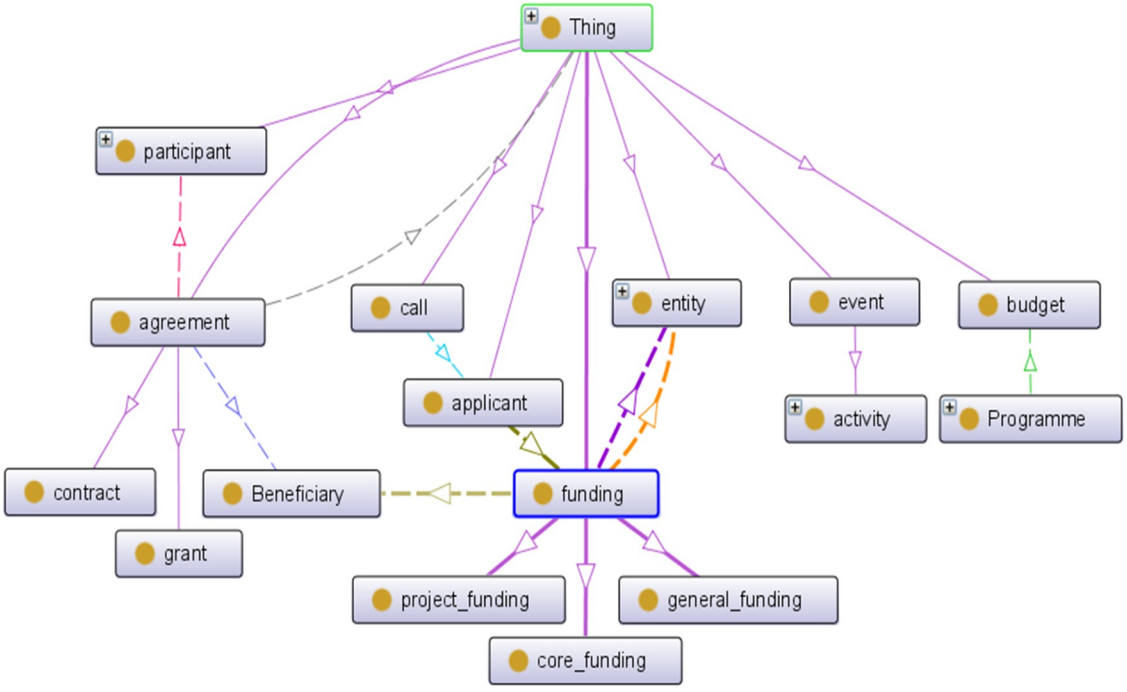

Fig. 3. A part of the INFUSE ontology.

the results. A 50/30/20 split is used to train, test, and validate the network's performance. The training was carried out till 75th epoch when the loss is minimum and as the network becomes stable. Since we were tackling a multi-class classification problem in which the dataset was not balanced, in this study we adopted and reported the most common performance metrics, namely weighted-average precision, recall, and F1 score, which take into account the class imbalance.

To determine how many neurons in each hidden layer of a 3-, 5-, and 7-hidden layer architecture gives best performance, the stopping criteria for training was set to 75 iterations. Table 2 shows the performance of MLP trained on single level of document representation using various number of hidden layers and different number of neurons per layer.

A significant improvement in the performance is observed as the number of neurons are increased in each layer for a 3-hidden

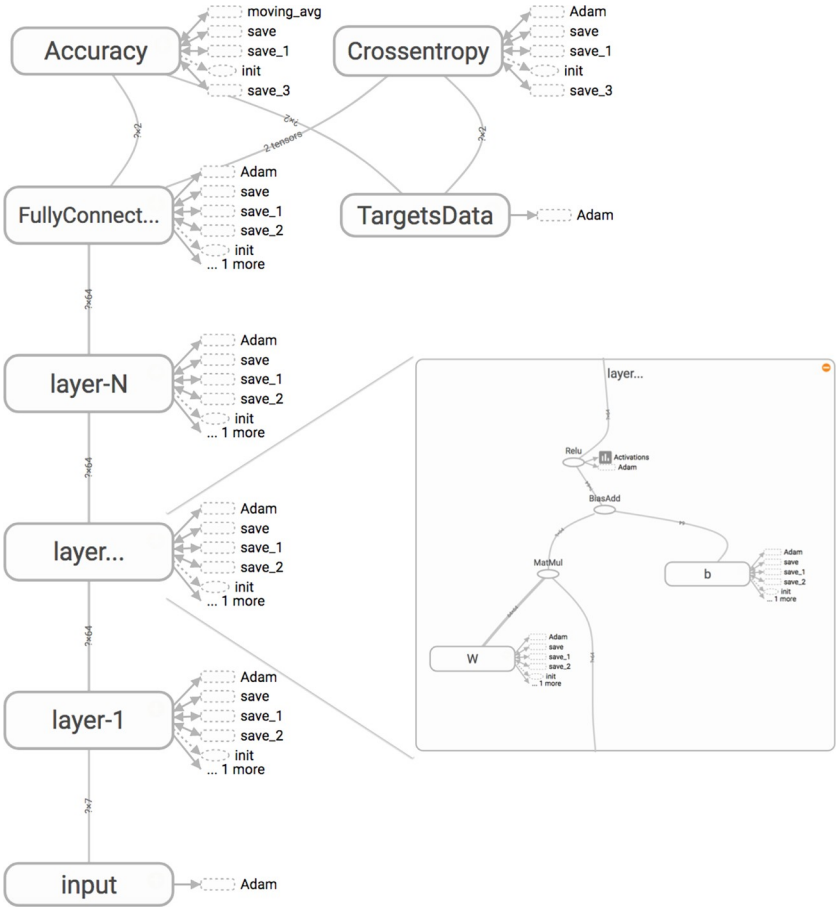

Fig. 4. A N-hidden layer fully-connected DNN model architecture with 64 neurons.

Table 1
Trainable parameters for a 5-hidden layer DNN with 64 neurons in each layer.

Layer (type)	Output shape	Param #
Hidden Layer_**1** (Dense)	(None, 64)	6464
Hidden Layer_2 (Dense)	(None, 64)	4160
Hidden Layer_**3** (Dense)	(None, 64)	4160
Hidden Layer_**4** (Dense)	(None, 64)	4160
Hidden Layer_**5** (Dense)	(None, 64)	4160
Output Layer_**6** (Dense)	(None, 5)	325
Total params: 23,429		
Trainable params: 23,429		
Non-trainable params: 0		

layer architecture. This holds true for the 5- and 7-hidden layer architecture as well. The increase in network performance when the number of neurons are increased is due to the fact that the input feature vector contains more than 100 attributes which are represented well when the network is trained with higher number of neurons. However, increasing the number of layers did not add much to the network's performance.

It was found that the 3-hidden layer MLP having 1024 neurons in each layer gave the best performance compared to the other MLP architecture configurations. The results are consistent with document representation enrichment using background knowledge derived by ontology (baseline ontology) and the relevant terminology (Top-N terms) attached to the ontology as shown in Table 3. Therefore, the other simulations for Top-1 to Top-5 terms as relevant terminology used for enriching semantically a document were carried out using only 3 hidden layers of network configuration having 1024 neurons in each layer.

Table 4 shows the overall results of the classification performance on the INFUSE dataset for 3-hidden layer MLP architecture with 1024 neurons in each layer. A significant improvement in the classification performance is achieved when documents are enriched with semantics using background knowledge provided by the ontology and its acquired relevant terminology. This is clearly evident

Table 2
Performance of MLP for single level of document representation (baseline).

# of layers	# of neurons	Precision (%)	Recall (%)	F1 score (%)
3-hidden	64	70.23	73.47	71.57
	128	72.87	73.47	72.20
	256	72.35	75.51	73.58
	512	72.45	75.51	73.57
	1024	72.59	75.51	73.82
5-hidden	64	67.18	69.39	67.77
	128	72.13	74.49	73.09
	256	74.95	77.55	75.80
	512	76.03	78.57	76.36
	1024	71.35	74.49	72.51
7-hidden	64	67.53	68.37	66.22
	128	68.11	71.43	69.27
	256	69.38	72.45	70.73
	512	71.02	74.49	71.99
	1024	72.63	75.51	73.86

Table 3
Performance of MLP for Top-5 levels of document representation.

# of layers	# of neurons	Precision (%)	Recall (%)	F1 score (%)
3-hidden	64	73.12	75.51	74.15
	128	75.89	78.57	77.10
	256	76.37	79.59	77.73
	512	76.41	78.57	77.36
	1024	76.77	79.59	**78.10**
5-hidden	64	72.98	73.47	73.09
	128	72.74	75.51	73.91
	256	73.83	76.53	74.98
	512	76.03	78.57	76.69
	1024	73.49	75.51	74.45
7-hidden	64	73.03	75.51	74.20
	128	71.80	74.49	72.76
	256	76.49	77.55	75.99
	512	72.35	74.49	73.05
	1024	74.09	73.47	73.61

Table 4
Performance of MLP on the INFUSE dataset for different levels of document representation [3 hidden layers, 1024 Neurons].

Doc representation	Precision (%)	Recall (%)	F1 score (%)
Baseline	72.59	75.51	73.82
Top-1 term	73.19	76.53	74.30
Top-2 terms	75.11	77.55	76.23
Top-3 terms	75.65	78.57	76.95
Top-4 terms	75.94	78.57	77.08
Top-5 terms	76.77	79.59	78.10

from Table 4 where a document enriched with Top-5 terms obtained a classification F1 score of 78.10%, almost 5 percentage points more than the document enriched using background knowledge gathered by only baseline ontology with only 73.82% F1 score.

To compare the performance of MLP with CNN and LSTM, we computed results for the 3-hidden layer architecture only with 1024 neurons in each layer for the other two models. Table 5 shows the architecture of a CNN with three convolutional layers and a global max pooling layer. The last dense layer is a fully connected softmax. The LSTM topology is kept the same as of CNN consisting of 3 hidden layers without the pooling layer.

Table 6 shows the performance of a 3-hidden layer CNN architecture on the INFUSE dataset for different levels of document representation. The results are similar to the MLP for documents enriched with semantics using only baseline ontology to documents enriched with semantics using baseline ontology and its relevant terminology attached to it.

Similar performance is obtained for LSTM as shown in Table 7. A constant improvement is seen for every point of testing as documents are enriched with semantics using baseline ontology and its relevant terminology integrated to it.

Though the performance of all three models shown in Tables 4, 6, and 7 increases significantly as the documents are enriched with semantics using baseline ontology and its relevant terminology integrated to it, but it is pretty evident that MLP gives better

Table 5
Trainable parameters for a 3-hidden layer CNN with 1024 neurons in each layer and a max pooling layer.

Layer (type)	Output Shape	Param #
Hidden Layer_**1** (Conv1D)	(None, None, 1024)	103424
Hidden Layer_*2* (Conv1D)	(None, None, 1024)	1049600
Hidden Layer_**3** (Conv1D)	(None, None, 1024)	1049600
Global_max_pooling1d_1	(None, 1024)	0
Output Layer_**1** (Dense)	(None, 5)	5125
Total params: 2,207,749		
Trainable params: 2,207,749		
Non-trainable params: 0		

Table 6
Performance of CNN on the INFUSE dataset for different levels of document representation [3 hidden layers, 1024 Neurons].

Doc representation	Precision (%)	Recall (%)	F1 score (%)
Baseline	65.44	68.37	66.45
Top-1 term	66.44	69.39	67.44
Top-2 terms	65.65	70.41	67.51
Top-3 terms	68.14	71.43	69.44
Top-4 terms	69.37	71.43	70.18
Top-5 terms	72.58	75.51	73.82

Table 7
Performance of LSTM on the INFUSE dataset for different levels of document representation [3 hidden layers, 1024 Neurons].

Doc representation	Precision (%)	Recall (%)	F1 score (%)
Baseline	61.00	62.24	60.61
Top-1 term	64.71	66.33	65.10
Top-2 terms	64.69	66.33	65.46
Top-3 terms	66.54	68.37	67.05
Top-4 terms	68.41	68.37	67.72
Top-5 terms	69.55	71.43	69.89

classification performance compared to the CNN and LSTM.

6.2. Results of DNN using VARIOUS document REPRESENTATION techniques

In this subsection, we conducted a comparative performance analysis of classification task by comparing our proposed document representation technique with three state-of-the-art representation techniques, namely term frequency inverse document frequency - *tf***idf*, word embedding, and topic modeling.

*tf***idf* is the simplest and most commonly used document representation technique which relies on distributional feature of words. It shows the relevance of words occurring in a document using words' local and global distributions. The former distribution known as term frequency *tf* reflects the importance of words in a document, while the later distribution called inverse document frequency *idf* shows the distribution of those words among the collection of documents.

A word embedding is a representation technique that employs dense vectors for word or document representations. These vectors are composed of continuous real values learned from text corpora and are of fixed sizes. Each word is associated with a value in the vector space. The value of the word is defined by words that accompany it and this allows to capture context in which words occur. The ability to capture context of words makes word embeddings more expressive representation technique. Furthermore, word embeddings allow through contextual similarity (cosine similarity distance) to capture complex syntactic and semantic relationships between words.

In this study, we generated a package of word embeddings with 300 dimensions. These embeddings are trained and learned on our corpus which comprised of 5.7 million words (tokens) with a vocabulary of 19,458 unique words. For obtaining unique words, we performed some pre-processing including removing all punctuation and capitalization, removing of stop words and words with length less than or equal to one character, and words that are not purely comprised of alphabetical characters.

Topic modeling is another technique that can be used for document representation. This technique relies on a generative statistical model which represents each document as a mixture of a small number of topics or themes. Each document topic is comprised of a topic-word distribution which is a distribution of words characterizing that topic. Grouping words with similar semantics into the document topics allow topic modeling technique to capture and exploit semantic relationships between words.

We learned a latent Dirichlet allocation (LDA) topic model with 300 topics corresponding to the dimensions of word embeddings generated from the INFUSE dataset. The same pre-processing steps are undertaken as for word embeddings.

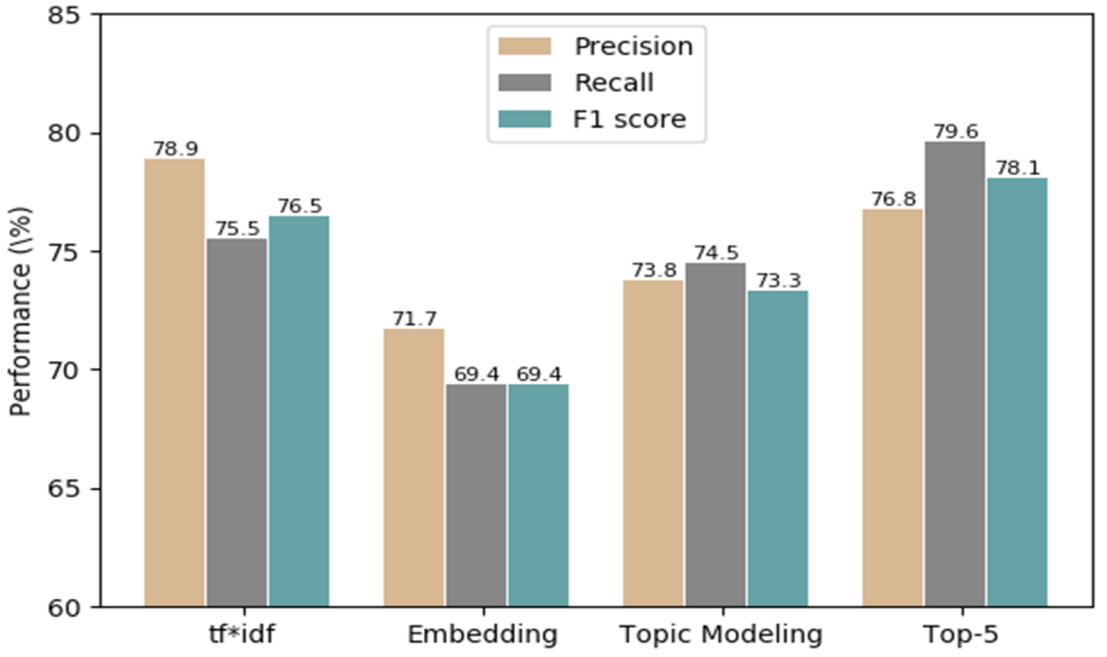

Fig. 5. Performance of MLP on the INFUSE dataset using different document representation techniques [3 hidden layers, 1024 Neurons].

Feature vectors generated from the three document representation techniques described above along with our proposed technique are used as input to train the MLP model and the obtained results are illustrated in Fig. 5. As can be seen from the diagram, MLP model using our document representation technique (denoted as Top-5 in diagram), achieves better performance in contrast to MLP using *tf*idf*, word embedding, and topic modeling, with 1.6%, 8.7%, and 4.8% F1 score improvement over them respectively. Having in mind that our representation technique employs only 323 feature vectors compared to 19,458 feature vectors which are used by other techniques (*tf*idf* and word embedding), makes it a very efficient and effective document representation technique.

6.3. Results of CONVENTIONAL MACHINE LEARNING techniques on the INFUSE DATASET

In this subsection, we provide results achieved by three different conventional machine learning (ML) techniques, namely Naive Bayes (NB), Support Vector Machine (SVM), and Decision Tree (DT).

Naive Bayes is a type of Bayesian network technique which relies on statistical-based learning model. The classification is done using Bayes rule with the assumption that all attributes of a dataset are independent to its class variable. The assumption of strong independence of the variables is the 'naive' part of this classification technique.

Naive Bayes can be considered as a parametric model because it can be parametrized by a fixed number of parameters. In essence, its statistical model is specified by a simplified function through a set of distributions given in the following Equation:

$$\mathcal{H} = p(x, w; \phi) \tag{7}$$

where, φ includes the parameter for the class prior probability $p\{w\}$, and the class conditional probability density function (posterior) $p\{x_i|w\}$ for each dimension. The conditional probability is parametric similar to continuous univariate normal density (Gaussian) or multivariate density.

On the contrary to Naive Bayes technique, Decision Tree is a nonparametric technique which assumes no prior parameterized knowledge about the underlying probability density function. In essence, the classification relies on the information provided by training samples alone.

Support Vector Machine is a classifier which can be either parametric or non-parametric model. Linear Support Vector Machine contains a fixed size of parameters represented by the weight coefficient. Therefore, it belongs to the family of parametric models. On the other side, non-linear Support Vector Machine is a non-parametric technique and Radial Basis Function Kernel Support Vector Machine, known as RBF Kernel SVM, is a typical example of this family. In a RBF Kernel SVM, it is the kernel matrix which makes it non-parametric. This kernel matrix is constructed by computing the pair-wise distances between the two feature vectors.

A Gaussian based Naive Bayes classifier is used in this paper for computing the results using a scikit[3] python package with default parameters. For SVM, an RBF kernel is used in which the value of gamma is set to 0.0001. This value indicates how much influence a single training sample has. The value of regularization parameter c is set to maximum. All other parameter values for conventional

[3] https://scikit-learn.org/stable/

Table 8
Performance score obtained by three different conventional ML techniques on the INFUSE dataset for different levels of document representation.

Model	Representation	Precision (%)	Recall (%)	F1 score (%)
NB	Baseline	52.54	54.08	51.07
	Top-1	58.01	59.18	57.52
	Top-2	64.32	62.24	62.73
	Top-3	64.07	62.24	62.49
	Top-4	61.32	60.20	60.25
	Top-5	62.01	61.22	61.28
SVM	Baseline	70.40	72.45	70.84
	Top-1	72.91	73.47	72.88
	Top-2	69.20	71.43	70.18
	Top-3	70.59	73.47	71.87
	Top-4	73.97	76.53	75.18
	Top-5	73.02	75.51	73.91
DT	Baseline	64.86	67.35	65.76
	Top-1	64.97	65.31	64.33
	Top-2	67.82	68.37	67.53
	Top-3	69.80	67.35	68.04
	Top-4	71.50	70.41	70.78
	Top-5	75.51	74.49	74.75

machine learning classifiers were set to default.

Table 8 shows a side by side comparison of classification performance obtained by the three conventional machine learning techniques on the INFUSE dataset. In this case, the documents are enriched with semantics using background knowledge provided by the ontology and the relevant terminology attached to it. As can be seen from the results shown in Table 8, a higher performance is achieved by all techniques when the classification is conducted using documents which are represented by baseline ontology and its relevant terminology comparing to the classification by using documents represented only by the baseline ontology. More concretely, Naive Bayes classifier has achieved an F1 score of 61.28% using documents enriched with baseline ontology and relevant terminology attached to it, compared to an F1 score of 51.07% using documents representation using only the baseline ontology. Almost the same increasing trend of performance is observed for SVM and Decision Tree classifiers which achieved an F1 score improvement from 70.84% to 73.91%, and 65.76% to 74.75%, respectively.

6.4. Comparison of DNN and conventional ML techniques

This subsection provides a comparison between DNN models and conventional ML techniques presented in Sections 6.1 and 6.3, respectively. The comparison is illustrated in the graph shown in Fig. 6. It can be seen from the chart diagram that the performance of all classifiers is significantly increased when documents are enriched with semantics using background knowledge provided by the ontology and the acquired relevant terminology attached to the ontology. The best classification performance in this case is achieved when the Top-5 terms are used for document representation. The obtained results suggest that acquisition of the relevant terminology

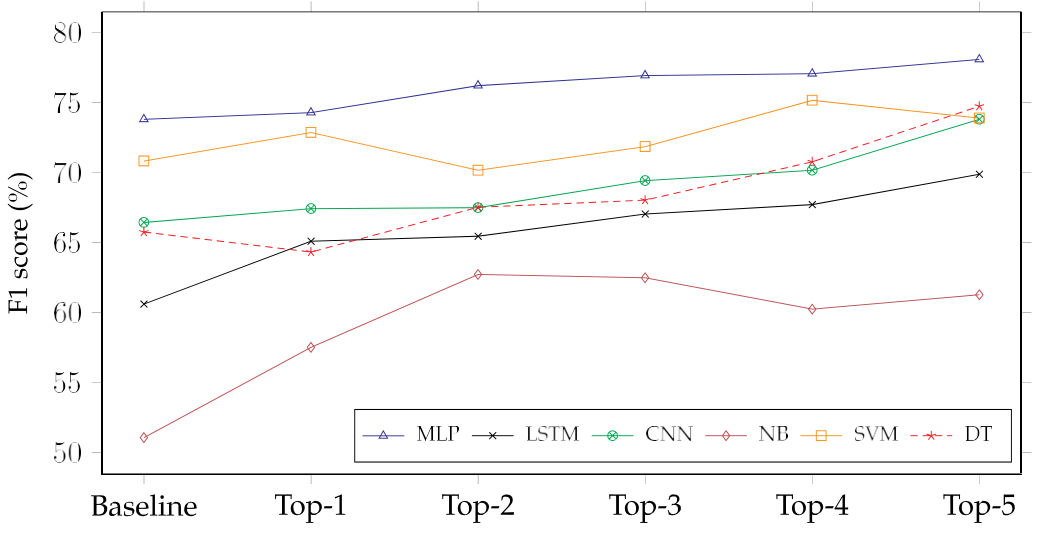

Fig. 6. F1 score achieved by DNN and conventional ML techniques on the INFUSE dataset for different levels of document representation.

attached to the ontology would improve the classification performance, regardless of the classification techniques employed.

It is also interesting to note from Fig. 6 that in general, DNN models outperform the conventional techniques. In particular, MLP yields the best performance among all the techniques. The worst performance is shown by NB and that may have happened due to the 'naive' property of the classifier which assumes that features are absolutely independent to each other.

Another interesting fact that can be drawn from the graph shown in Fig. 6 is that MLP from DNN, and DT and NB from conventional ML achieve completely opposite results when it comes to using Top-2 terms level of document representation as compared to LSTM and CNN from DNN models, and SVM from ML techniques, respectively. This may happen due to the fact that LSTM is capable of preserving the dependency over long periods of time and this may generate some noisy relationships between attributes that in turn can degrade the classification performance. In the same fashion, SVM is very sensitive to outliers which could account for performance degradation.

7. Conclusion and future work

This paper presented a real-case example of ontology-based document classification from funding domain on the INFUSE dataset. The classification system relies on semantically rich document representation achieved by using background knowledge provided by ontologies and the relevant terminology integrated to them which is extracted by aggregating semantic and contextual information. Semantic representation of a document is then used as the input data to the deep learning classifier for assigning that document to the appropriate category.

The proposed classification system is evaluated on three deep neural network architectures on multiple configurations of the network. The results on the real-life ontology are compared to those obtained with conventional machine learning classifiers. Deep learning classifiers showed an average improvement of 4% points over the conventional ML classifiers on the INFUSE dataset. The findings also revealed that by associating relevant semantic terms to the documents significantly increases the classification performance. On conventional classifiers, the increase was about 3%-10% points between baseline to Top-5 terms level of document representation, while in case of deep learning architecture, the improvement was around 4%–9% points. This also shows that the deep learning classifiers can better represent and classify the documents even for baseline document representation in comparison to conventional ML classifiers. Nevertheless, adding semantically relevant terms (levels of representation) to documents for classification helps improve the performance by at least 5 percentage points.

We also evaluated different combinations of the feed-forward network on the INFUSE dataset by varying the number of layers and number of neurons in each layer. It was found that increasing the neurons from 64 to 1024 significantly improves the classification performance with respect to F1 score, however, increasing the layers did not add much in terms of the classification performance due to the limited amount of training samples. Future work must, therefore, focus on adding more real-case financial documents to the INFUSE dataset to make deep learning more efficacious for enhancing network's performance. Additionally, using more datasets from other domains would certainly be of great interest for generalization of the proposed approach because our study has been limited to use a single dataset due to the difficulty of collecting and securing text corpora and the ontology.

References

Agarwal, B., Ramampiaro, H., Langseth, H., & Ruocco, M. (2018). A deep network model for paraphrase detection in short text messages. *Information Processing and Management, 54*(6), 922–937.

Altinel, B., & Ganiz, M. C. (2018). Semantic text classification: A survey of past and recent advances. *Information Processing and Management, 54*, 1129–1153.

Bing, L., Jiang, S., Lam, W., Zhang, Y., & Jameel, S. (2015). Adaptive concept resolution for document representation and its applications in text mining. *Knowledge-Based Systems, 74*, 1–13.

Cagliero, L., & Garza, P. (2013). Improving classification models with taxonomy information. *Data & Knowledge Engineering, 86*, 85–101.

Camous, F., Blott, S., & Smeaton, A. F. (2007). *Ontology-based MEDLINE document classification. Proceedings of the International conference on bioinformatics research and development*439–452.

Dinh, D., & Tamine, L. (2011). *Biomedical concept extraction based on combining the content-based and word order similarities. Proceedings of the ACM symposium on applied computing*1159–1163.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning.* MIT Press.

Hassan, A., & Mahmood, A. (2017). *Deep learning for sentence classification. Proceedings of the IEEE international conference on long island systems, applications and technology*1–5.

Hassan, A., & Mahmood, A. (2018). Convolutional recurrent deep learning model for sentence classification. *IEEE Access, 6*, 13949–13957.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.

Kastrati, Z., Imran, A. S., & Yayilgan, S. Y. (2015). *An improved concept vector space model for ontology based classification. Proceedings of the International conference on signal image technology & internet systems*240–245.

Kastrati, Z., Imran, A. S., & Yayilgan, S. Y. (2016). SEMCON - A semantic and contextual objective metric for enriching domain ontology concepts. *International Journal on Semantic Web and Information Systems, 12*(2), 1–24.

Kastrati, Z., & Yayilgan, S. Y. (2017). *Supervised ontology-based document classification model. Proceedings of the International conference on compute and data analysis*245–251.

Keikha, M., Khonsari, A., & Oroumchian, F. (2009). Rich document representation and classification: An analysis. *Knowledge-Based Systems, 22*(1), 67–71.

Kim, H. J., Kim, J., Kim, J., & Lim, P. (2018). Towards perfect text classification with wikipedia-based semantic naive bayes learning. *Neurocomputing, 315*, 128–134.

Kim, Y. (2014). *Convolutional neural networks for sentence classification. Proceedings of the International conference on empirical methods in natural language processing (emnlp)*1746–1751.

Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., & Barnes, L. E. (2017). *Hdltex: Hierarchical deep learning for text classification. Proceedings of the IEEE International conference on machine learning and applications*364–371.

Miller, F. P., Vandome, A. F., & McBrewster, J. (2009). *Levenshtein distance: Information theory, computer science, string (computer science), string metric, Damerau?Levenshtein distance, spell checker, Hamming distance.* Alpha Press.

Pak, A. N., & Chung, C.-W. (2010). A wikipedia matching approach to contextual advertising. *World Wide Web, 13*(3), 251–274.

Phan, N., Dou, D., Wang, H., Kil, D., & Piniewski, B. (2017). Ontology-based deep learning for human behavior prediction with explanations in health social networks.

Information Sciences, 384, 298–313.
Sanchez-Pi, N., Marti, L., & Garcia, A. C. B. (2014). *Text classification techniques in oil industry applications. Proceedings of the international joint conference SOCO'13- CISIS'13-ICEUTE'13*211–220.
Sanchez-Pi, N., Marti, L., & Garcia, A. C. B. (2016). Improving ontology-based text classification: An occupational health and security application. *Journal of Applied Logic, 17*, 48–58.
Serban, O., Thapen, N., Maginnis, B., Hankin, C., & Foot, V. (2018). Real-time processing of social media with sentinel: A syndromic surveillance system incorporating deep learning for health classification. *Information Processing and Management,* 1–19.
Sy, M.-F., Ranwez, S., Montmain, J., Regnault, A., Crampes, M., & Ranwez, V. (2012). User centered and ontology based information retrieval system for life sciences. *BMC Bioinformatics, 13*(1), 1–12.
Wei, D., Wang, B., Lin, G., Liu, D., Dong, Z., Liu, H., et al. (2018). Research on unstructured text data mining and fault classification based on rnn-lstm with malfunction inspection report. *Energies, 10*(3), 1–22.
Wu, Z., & Palmer, M. (1994). *Verbs semantics and lexical selection. Proceedings of the annual meeting on association for computational linguistics*133–138.
Wu, Z., Zhu, H., Li, G., Cui, Z., Huang, H., Li, J., et al. (2017). An efficient wikipedia semantic matching approach to text document classification. *Information Sciences, 393*(C), 15–28.
Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). *Hierarchical attention networks for document classification. Proceedings of the conference of the north american chapter of the association for computational linguistics: Human language technologies*1–10.
Zhang, W., Du, Y., Yoshida, T., & Wang, Q. (2018). Dri-rcnn: An approach to deceptive review identification using recurrent convolutional neural network. *Information Processing and Management, 54*, 576–592.
Zhang, X., Zhao, J., & LeCun, Y. (2015). *Character-level convolutional networks for text classification. Proceedings of the international conference on advances in neural information processing systems*649–657.