


# Knowledge-Infused Text Classification for the Biomedical Domain

Sonika Malik, Maharaja Surajmal Institute of Technology, India\*

Sarika Jain, National Institute of Technology, Kurukshetra, India

 <https://orcid.org/0000-0002-7432-8506>

## ABSTRACT

Extracting knowledge from unstructured text and then classifying it is gaining importance after the data explosion on the web. The traditional text classification approaches are becoming ubiquitous, but the hybrid of semantic knowledge representation with statistical techniques can be more promising. The developed method attempts to fabricate neural networks to expedite and improve the simulation of ontology-based classification. This paper weighs upon the accurate results between the ontology-based text classification and traditional classification based on the artificial neural network (ANN) using distinguished parameters such as accuracy, precision, etc. The experimental analysis shows that the proposed findings are substantially better than the conventional text classification, taking the course of action into account. The authors also ran tests to compare the results of the proposed research model with one of the latest researches, resulting in a cut above accuracy and F1 score of the proposed model for various experiments performed at the different number of hidden layers and neurons.

## KEYWORDS

Artificial Neural Network, Classifier, Natural Language Processing, Ontology, Text Classification

## 1. INTRODUCTION

The classification of disease based on the symptoms provided as input proves to be a challenging task, which can be simplified using the machine learning algorithm. Text classification is a field that holds the enormous capability to classify text, but it remains a difficult task if done manually (Korde & Mahender, 2012). Text mining has various applications like classification (Supervised, Semi-supervised, Unsupervised), sentiment analysis, and document filtering. Machine Learning and Natural Language Processing approach collaborate to find the patterns in various documents and automatically classify them.

Artificial intelligence (AI) is expected to generate hundreds of billions of dollars in economic value. Although technology has become a part of our daily life, many individuals are still suspicious. It's a major problem for them as AI approaches work like black boxes and create ideas mysteriously. Furthermore, several sectors have accepted knowledge graphs (KGs) as a useful tool for data

DOI: 10.4018/IJISMD.306635

\*Corresponding Author

This article published as an Open Access Article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

administration, processing, and enrichment (Jain, 2021). In spite of this, KGs are increasingly being recognized as the foundations of an AI system that provides explainable AI through the design concept known as “Human in the Loop” (HITL). The AI’s promise is to automatically derive patterns and rules from large datasets using machine learning algorithms like deep learning. In many circumstances, this makes categorization activities much easier to perform. While machine learning algorithms can collect knowledge from historical data, they are unable to produce new results using that knowledge. There is no trust if nothing is explained. Explain ability ensures that trustworthy agents in the system can understand and defend the AI agent’s decisions. Symbolic AI and statistical AI are combined in semantic AI. It incorporates methods such as machine learning, information analysis, semantic web mining, and text mining. It blends neural networks and semantic reasoning with AI techniques. By using this new framework, AI-based systems can be built more quickly and efficiently (Malik & Jain, n.d.).

There’s a traditional way of presenting documents called Bag of Words (BOW). Using this strategy, you get information about the terms and their respective frequencies within the sentence or document. Due to each document being represented as a vector of term frequencies in the lexicon, it is also known as the Vector Space Model (or VSM). The semantic relationships between words are also ignored in this representation (Salton & Yang, 1973). The words are also completely out of sequence when they are represented by BOW. That’s because this strategy stresses that phrases have some frequency information attached to them.

On the other hand, the task of a text classifier is to classify textual documents according to predetermined categories, with the apparent assumption of each class consisting of records with similar content, usually discussing a particular topic that is different from other classes. When texts are displayed in a vector space, they tend to be sparse due to high dimensionality (Altinel & Ganiz, 2018). This is a significant challenge, especially when there are a large number of class labels but insufficient training data for each of them. If you’re working with real-world applications, obtaining labelled data of high quality for training is frequently prohibitively expensive. It is, therefore, necessary to have an accurate text classifier that can make use of semantic information.

There are a variety of techniques for classifying documents based on their semantics. It is based on the meaning of words and hidden semantic relationships between words and documents. Semantic text classification has some advantages over standard text categorization:

- Relationships between words might be implied or explicit.
- Word-to-document correlations are extracted and used.
- Possibility of generating keyword representations for existing classes.
- Classification accuracy is improved.
- Traditional text classification algorithms cannot manage synonymy and polysemy because they do not take into account semantic links between words.

In this paper, we have shown that machine learning with ontology gives better accuracy than traditional text classification with the same dataset. We also compared our results with one of the latest researches and got better accuracy and F1-score (Althubaiti et al., 2020). The authors used ANN for ontology-based text classification with the same ontology and dataset. They have used Machine learning and word embeddings to identify words and phrases that refer to an ontology class. They used word embeddings to collect the context information and relied on the automatic feature selection technique and ANN as a classifier. One of the drawbacks in automatic feature ranking is that they are univariate: each feature is considered separately, ignoring dependencies between features. In this biological domain, free text is matched with the ontology concepts that encode the semantic relationships. According to our hypothesis, the dependency between concepts stored in biological ontologies could be used to improve the feature ranking (Garla & Brandt, 2012).

This paper aims to extend the classification theory using neural networks to get better accuracy with ontology. We have used Keras for the Neural Network model and trained it on a dataset extracted

from UMLS, an official website for Medicinal researched. Using Keras, we have designed the different hidden layers, which help to improve the accuracy of text classification. A Feedforward Neural network is used for the code, which has the characteristic that the output of one layer is feed to the next layer.

The main contributions of this work are as follows:

- To compare traditional text classification with ontology-based text classification.
- To propose an algorithm for ontology-based text classification using an artificial neural network for better accuracy.

The rest of the paper is organized as follows.

Section 2 describes the Literature review on text classification. In Section 3, we discuss the existing techniques for text classification. Section 4 describes the detailed description of our proposed work. In Section 5, we evaluate the proposed approach against the existing one. Section 6 represents the threats of validity. Finally, Section 7 concludes the paper.

## 2. LITERATURE REVIEW

The main intention of this paper is to explore and present chronologically, a comprehensive survey of the major applications of deep learning covering a variety of areas, the study of the techniques and architectures used, and further the contribution of that respective application in the real world. Finally, the paper ends with the conclusion and future aspects.

The traditional text classification approaches use Machine Learning or Statistical methods to perform any task. Such methods include Naïve Bayes (Lewis, 1998), Support Vector Machines (Vapnik, 2013), Latent Semantic Analysis (Deerwester et al., 1990), and many others. The Classical text classification methods are provided clearly and concisely in (Sebastiani, 2002). In (Dargan et al., 2020), the authors focus on the concepts of deep learning and its applications covering various areas, study the techniques and architectures used, and further the contribution of that application in the real world. The authors in (Dagan et al., 1997; Kaur & Sapra, 2013; Ruiz & Srinivasan, 1997; Wu & Tsai, 2009) presented different Neural Networks for text classification on different applications like spam filtering, document classification, etc. The authors in (Ng et al., 1997) developed a new feature selection method to reduce dimensionality and ease classification. In a lot of these researches, textual and numeric data have been used to improve the classification and prediction of these expert systems (Arbelaez et al., 2012; Fageeri et al., 2017; Waudby et al., 2011; WHO, 2010). Automatic text classification is the process of allocating a text to one or more pre-defined classifications based on its content. Many applications use text classification techniques, including monitoring of news and its classification, e-mail filtering, sorting through digitized paper archives, automated indexing of scientific articles, and finding interesting information on the Web, as well as biomedical applications (De Paz et al., 2013). A comprehensive survey of hybrid classifier systems can be found in (Woźniak et al., 2014).

However, there is still much work to be done on incorporating semantic background knowledge into text categorization. Early several authors have worked on it and some are: (Bodner & Song, 1996; Dave et al., 2003; Green, 1999). To improve the text clustering task, they apply WordNet (Fellbaum, 1999). WordNet is a collection of related words grouped into synonym sets, with each set representing a lexical concept. WordNet has been used for text categorization and clustering with great success. In (Ma et al., 2012), the authors developed an ontology for classifying research proposals. There are four stages: (a) Constructing research ontology, (b) Classifying new research proposals into disciplines, (c) building research proposal clusters using text mining, (d) balancing research proposals and regrouping them by considering applicants' characteristics. The authors in (Kaur & Sapra, 2013) researched in a similar domain; they proposed Ontology-Based text mining methods to classify research proposals and external research reviewers. The authors in (Sanchez-Pi et

al., 2014) proposed a classification strategy for classifying oil and gas sector incidents that relied on prior knowledge provided by an ontology. It is worth noting that the basic knowledge provided by the domain ontology for health, safety and environmental contexts for oil and gas applications has been extended to include a thesaurus to find implicit relationships to make the approach more flexible and resistant to the classification of real documents that can be written in the heterogeneous form. The authors in (Sanchez-Pi et al., 2016) resented an extension of this classification approach which, in addition to the basic knowledge gained from ontology, uses a list of technical terms generated semi-automatically using n-gram extraction method. In (Althubaiti et al., 2020), authors have developed a framework for identifying terms and phrases used in biomedical Europe PMC full-text papers for a class in ontology. To this end, the authors define variants of the lexical representation, use word embedding to collect information from meaning, and use online reasoning to generate features and an artificial neural network to classify them.

### 3. BACKGROUND KNOWLEDGE

This section contains and handles the theoretical background needed to solve and understand the problem area. We will also discuss the techniques of text classification.

#### 3.1 Machine Learning Techniques for Text Classification

The ability of a computer to learn from data or previous experience is referred to as Machine Learning (ML). The traditional text classification is shown in Figure 1.

There are three primary forms of machine learning:

- **Supervised ML:** The learning process is guided by human assistance in the form of labelled training data.
- **Unsupervised ML:** There is no human involvement. The classifier uses unlabelled training data to find some unknown categories into which the data can be clustered.
- **Semi-supervised ML:** Labelling only a small portion of the training data provides only partial human assistance.

However, supervised ML techniques need more manual effort to prepare the training data than unsupervised and semi-supervised methods. Still, they often produce better results due to the extra-human assistance (Sebastiani, 2002). Some commonly used supervised and semi-supervised ML algorithms include SVM, Naïve Bayes (NB), k-nearest neighbor's (k-NN), and decision trees (DT), etc. (Agarwal et al., 2011). When it comes to machine learning, SVM is the most often used algorithm. A feature space maps the labelled data and tries to discover the optimal separators between the data. After that, the test results are then mapped to feature space and classified by the founding separators. The k-Means and hierarchical algorithms are two unsupervised ML techniques that are frequently employed.

Deep learning and shallow learning are the two main types of machine learning. Shallow learning is limited to a linear combination of parameters and a single function from the training data. Text Classification has had success with shallow learning algorithms (e.g., SVM). Still, their

Figure 1. Traditional Text classification



poor modelling and representational power prevent them from learning complex functions such as those involved in text semantics. On the other hand, from training data based on deep learning, complicated functions can be learned with nonlinear combinations of parameters (Bengio & LeCun, 2007). The neural network algorithm is the most often used deep learning algorithm. An algorithm can simulate the iterative learning process of the human brain using neural networks, which learn from known information and infers new unknown information based on the learned knowledge. The feedforward neural network algorithm and the recurrent neural network algorithm are two examples of neural network algorithms that can be used. The hierarchical SoftMax skip-gram algorithm is the most up-to-date and best-performing algorithm.

In the traditional text classification basically, there are three main steps:

- Dataset Generation
- Model Training and Testing
- Analyzing/classifying results

### 3.2 Ontology-Based Techniques for Semantic Text Classification

To facilitate Text Classification, Semantic Text Classification makes use of the semantics of the text. When we talk about ontologies, we're talking about knowledge conceptualization. As a result, an ontology can help capture the text's meaning.

Document classification may benefit by using ontologies in the following two ways:

1. Use an ontology to represent a document's features and then use a machine learning method to categorize documents based on their features. For example, Term features from documents are extracted and mapped to the ontology concepts (Lee et al., 2008). Term features are replaced with ontology concept features in documents that used to be represented by term features. After that, ML-based TC is applied to these concept features.
2. Use the concept features of each category to classify documents based on semantic similarity scores between concepts or phrases. For example, in (Yu et al., 2006), the authors use a linguistic ontology with statistical data (such as word frequency) for text classification.

Concepts that describe syntactic and semantic features of words are included in the ontology's scope. These syntactic and semantic features are learned using a collection of labelled training data. Documents' keywords are extracted and classified using TC's linguistic ontology knowledge. The authors use concept vectors for text classification in (Yang et al., 2008). An ontology-derived concept is represented as a vector of concept-value pairs, where the value is defined based on the term frequency-inverse document frequency (TF-IDF). One way to express a testing document is by using a vector of keyword-TFIDF pairs. Based on the similarity between document vectors and category vectors, the documents are subsequently classified. Instead of requiring labelled training data like in the first example, the second example can be classified as an unsupervised ontology-based attempt. Unsupervised ontology-based TC, as opposed to supervised ML-based TC, eliminates the need to classify training data manually.

## 4. KNOWLEDGE-INFUSED TEXT CLASSIFICATION

The classification by neural network is supposed to improve with the use of ontology in the classification process. We aim to verify this fact by studying and comparing values of accuracy, precision, recall, and F1 score for ontology-based classification and traditional text classification.

In Ontology-based text classification, the second phase is being added between Dataset generation and Model training and testing, which is ontology matching, as shown in Figure 4 (a). The phases are:

(a) Dataset Generation, (b) Ontology Matching, (c) Model training and testing, and (d) Analysing/ classifying results:

1. **Dataset Generation:** Although a disease-symptom association database was already accessed by some authors with three columns for disease name, number of occurrences, and symptoms (Columbia, n.d.); however, it needed to be modified in order to be used for our research. Some other information was also included in order for a more precise matching process. It was therefore decided to use the final dataset that was developed for the intended research. The disease name is a classification/output feature in the modified dataset, while the disease description is a matching feature in the ontology. the ontology used is Human Disease Ontology, which was hosted at the website of the Institute of Genome Sciences, Maryland School of Medicine. It uses prefixes as identifiers for identifying diseases. Each prefix begins with 'DOID', followed by some number, which is used as disease identifiers. After the ontology and a workable dataset were obtained, the dataset was cleaned and pre-processed before being processed with Natural Language Toolkit (NLTK). The basic steps of Pre-processing are tokenization, stemming, lemmatization, and POS tagging. Frequency Distribution is applied to our filtered dataset in the end to obtain the frequency of the words in the diseases and symptoms our model will process. A list of tuples describing the keyword, frequency, and also whether the word is being lemmatized or not in the cleaning process is used. The lemmatizing factor (0 or 1) helps to sort the list according to their possibility of lemmatization keeping the non- lemmatized word in the start and later at the end, thereby helping in better class detection of possible disease. In addition, a synthetic dataset is created by employing computer techniques to create fresh data.
2. **Ontology Matching:** In this phase, the keywords generated from the disease definition match the keywords of ontology nodes. To construct the subset of data, all the matching nodes are possible classes for efficient model training. The use of priorities helps us restrict classes further. Every keyword is allocated two numbers in our ontology research to give priority to each keyword. The first number defines the keyword frequency, and the second number indicates whether or not the keyword can be lemmatized. It is assigned as 1 otherwise if it cannot be lemmatized. Each keyword, therefore, has a syntax (name, priority number, second priority number). Figure 4 (b) shows the steps for ontology matching.
3. **Model Training and Testing:** In this phase, we need a dataset and train a classifier on it using ANN. Because the dataset initially contains text keywords, the count vectorizer module must transform it to numbers. A classifier can then be trained using the training data. The neural network used is Keras Sequential Model (Abrori, 2016) as shown in Figure 2. Now to run the neural network, and observe the classification parameters for both the cases considered in the research. Before feeding the train and test data to the model, it is first pre-processed using Count Vectorizer, Label Encoder, and OneHotEncoder:
  - a. **Count Vectorizer:** Applied on the train and test data to obtain a sparse matrix of the vocabulary along with the respective frequency of tokens. The sparse matrix is created using the `csr_matrix` of the sklearn library. The maximum features taken in this research are 300.
  - b. **Label Encoder:** Used for the train and test labels to assign a number to the resultant labels which are diseases in our case.
  - c. **One Hot Encoder:** After performing Label Encoding, the labels are passed to OneHotEncoder to create a matrix that is passed to the neural network.

The model summary of the neural network is given in Figure 3.

In the neural network, we have one input layer containing 300 input neurons for each of the input features. Three hidden layers are consisting of 32 neurons each result of which is fed to the output layer which gives the predicted labels. Categorical Cross entropy is the loss function used in our study as given in Eq. 1. It is also called SoftMax loss which is a combination of SoftMax activation and

Figure 2. Keras Sequential Model (Abrori, 2016)

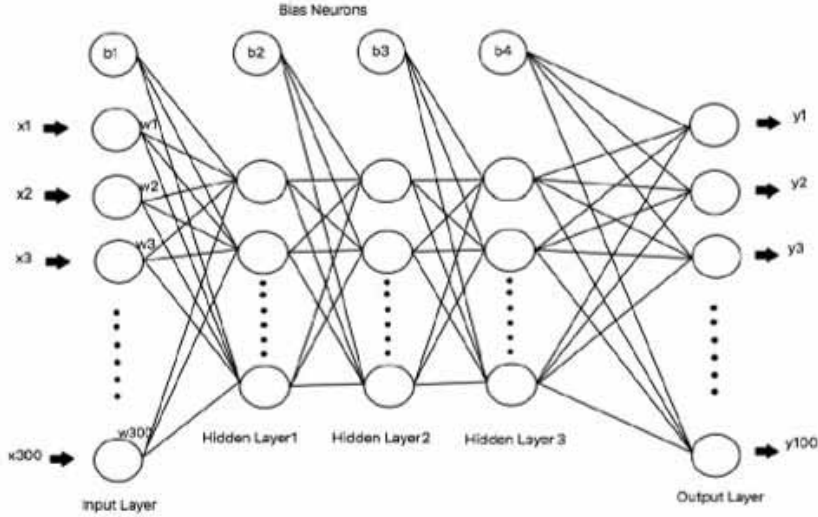


Figure 3. Model Summary

Model: "sequential\_26"

Layer (type)	Output Shape	Param #
dense_130 (Dense)	(None, 300)	90300
dense_131 (Dense)	(None, 32)	9632
dense_132 (Dense)	(None, 32)	1056
dense_133 (Dense)	(None, 32)	1056
dense_134 (Dense)	(None, 130)	4290
Total params: 106,334		
Trainable params: 106,334		
Non-trainable params: 0		

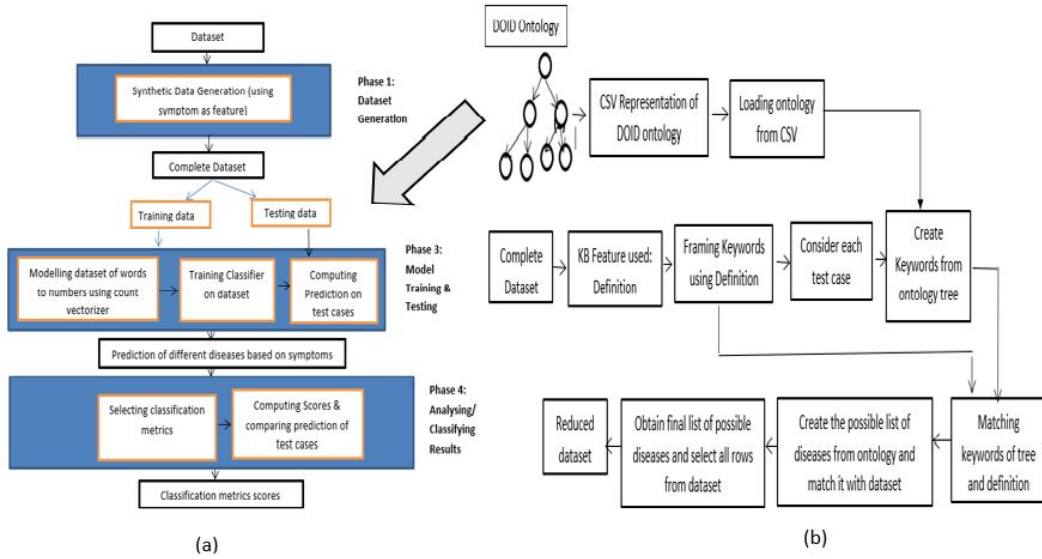
a cross-entropy loss which after training the neural network outputs the probability over the classes for each test instance:

$$L(y, \hat{y}) = - \sum_{j=0}^M \sum_{i=0}^N (y_{ij} * \log(\hat{y}_{ij})) \quad (1)$$

The result thus obtained is compared with the test labels to determine the accuracy, precision, recall, and F-measure and compare the results with the non-ontology-based approach. After training, the model can be used to make predictions based on the testing data. The ratio between training and testing data is 80/20.

4. **Analysing/Classifying Results:** For the result analysis, the disease predictions for the testing data are compared to the actual disease class. We calculated classification metrics such as accuracy, precision, recall, and F1-score after comparing the results. We have compared the performance

Figure 4. (a) Ontology-Based Text Classification 4; (b) Ontology Matching



of an Artificial Neural Network using metrics after this computation. On the basis of individual class-wise precision and recall values, we can also verify which classes appear to perform well.

## 5. RESULTS AND DISCUSSION

The following Parameters are used for evaluation and comparison of the model are accuracy, precision, recall, and F1 score:

- **Accuracy:** Accuracy defines the intuitive essence of a training model. The correctly predicted findings in the list of all predictions are taken into consideration:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- **Precision:** Precision is defined as the pertinent fraction of all instances found. It is also called the positive predicted value:

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:** Recall is the proportion of the instances found in all relevant instances. It is also called sensitivity:

$$Recall = \frac{TP}{TP + FN}$$

- **F1 Score:** Score is the weighted average of Precision and Recall. Therefore, it factors in false positives as well as false negatives. In case of an uneven class distribution, F1-score becomes more important than accuracy. Other times, when false negatives and positives have the same cost, accuracy may be treated as the superior evaluation parameter:

$$F1\ Score = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

Here:

*TP = True Positive*

*TN = True Negative*

*FP = False Positive*

*FN = False Negative*

## 5.1 Comparison of Traditional and Ontology-Based Text Classification

We have done the comparison of with and without ontology-based text classification using Machine Learning. The following are the results in Table 1. It's worth noting that the decision tree classifier has the largest boost in accuracy, precision, recall, and F1 score, with 0.75 for simple classification, and 0.85 for ontology-based classification, as well. The decision tree classifier has a 10% improvement in metrics for 500 test cases and a 6% percent improvement for 100 test cases. KNN Classifier, which exhibits a 5% improvement for both 100 and 500 test instances, is the next step in the process of learning. There has been a slight improvement in metrics for the rest of the classifiers as seen in Table 1.

In basic text classification and ontology-based classification, the parameter values for the bagging classifier are 0.87 and 0.89, respectively. For each classification, the parameters of the Random Forest Classifier were found to be 0.96 and 0.97. It's worth noting that for simple text classification, Nave-Bayes Classifier and SVM Classifier have the same values of all parameters as 0.98 and 1.0 for the ontology-based classifier, respectively. We'll now look into Logistic Regression, with metrics of 0.99 for simple text classification and 1.0 for ontology-based text classification. On the other hand, the order in which various classifiers improve their accuracy (with or without employing ontology). The graphical representation for 100 test cases is shown in Figure 5.

The following is the order of the classifier in relation to its magnitude:

Decision Tree Classifier < KNN Classifier < Bagging Classifier < Random Forest Classifier < Naïve Bayes Classifier < SVM classifier < Logistic Regression

### 5.1.1 Discussion

The findings reveal that the ontology-based classification is better than the classification that does not use ontologies. The general pattern points to an ontology-based classification that is more accurate and precise. When classification was done with the help of ontology, all of the measures employed (accuracy, precision, recall, and F1 Score) showed an increase of up to 10%. It may be concluded that the use of an ontology improved classification efficiency. In reference to Table 1, the values of the metrics Accuracy, Precision, recall, and F1-score have the same magnitude. This is due to the fact that the FP & FN values are the same in magnitude as there is a smaller number of records per disease. This benefit can be linked to the fact that the number of possible categorization classes has decreased, which has resulted in a reduction in the amount of time required for training.

Table 1. Comparison of with and without ontology-based text classification

Classifier	Parameter	100 Test Cases		500 Test Cases	
		Without Ontology	With Ontology	Without Ontology	With Ontology
Naïve-Bayes	Accuracy	0.98	1.0	0.97	0.978
	Precision	0.98	1.0	0.97	0.978
	Recall	0.98	1.0	0.97	0.978
	F1 Score	0.98	1.0	0.97	0.978
Decision Tree	Accuracy	0.74	0.80	0.758	0.852
	Precision	0.74	0.80	0.758	0.852
	Recall	0.74	0.80	0.758	0.852
	F1 Score	0.74	0.80	0.758	0.852
KNN	Accuracy	0.81	0.86	0.818	0.854
	Precision	0.81	0.86	0.818	0.854
	Recall	0.81	0.86	0.818	0.854
	F1 Score	0.81	0.86	0.818	0.854
Random Forest	Accuracy	0.96	0.97	0.932	0.952
	Precision	0.96	0.97	0.932	0.952
	Recall	0.96	0.97	0.932	0.952
	F1 Score	0.96	0.97	0.932	0.952
SVM	Accuracy	0.98	1.0	0.986	0.992
	Precision	0.98	1.0	0.986	0.992
	Recall	0.98	1.0	0.986	0.992
	F1 Score	0.98	1.0	0.986	0.992
Bagging	Accuracy	0.87	0.89	0.894	0.912
	Precision	0.87	0.89	0.894	0.912
	Recall	0.87	0.89	0.894	0.912
	F1 Score	0.87	0.89	0.894	0.912
Logistic Regression	Accuracy	0.99	1.0	0.982	0.99
	Precision	0.99	1.0	0.982	0.99
	Recall	0.99	1.0	0.982	0.99
	F1 Score	0.99	1.0	0.982	0.99

## 5.2 Ontology-Based Text Classification Using ANN

We have taken 4 experiments on different hidden layers 10, 50, 100 with 4 different classifications of diseases as shown in Table 2. In disease classification, there are 2 classes and the authors have given the highest accuracy as 96.06% and F1 score as 95.32%, while in our experiment we have got the accuracy and F1 score as 100%. Similarly, in the infectious disease classification, there are 5 classes, the highest accuracy, and F1 score is 95.74 % and 96.01% respectively, while we have got highest one is 100%. Next in the case of Anatomical disease, there are 13 classes and the highest accuracy and F1-score are 76.98 % and 70.20% respectively. In the last case, there is the combination of Infectious

Figure 5. Graphical Representation of Traditional and Ontology-based Text Classification

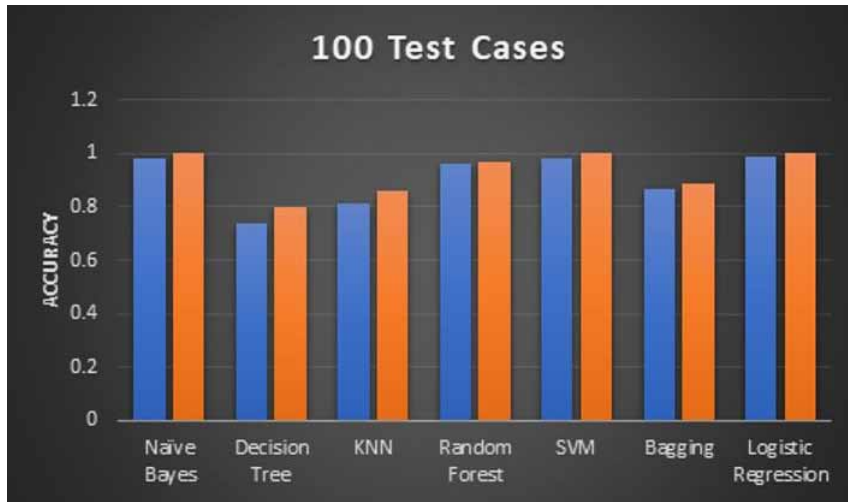


Table 2. Comparison of F1-Score and AUC for different classes using different hidden layers

Classification	Existing Approach (Fellbaum, 1999)							Proposed Approach											
								Without Ontology						With Ontology					
	Hidden Layers	10		50		100		10		50		100		10		50		100	
		No. of Classes	F1-Score	AUC	F1-Score	AUC	F1-Score	AUC	F1-Score	AUC	F1-Score	AUC	F1-Score	AUC	F1-Score	AUC	F1-Score	AUC	F1-Score
Diseases	2	94.65	95.31	94.83	95.97	<b>95.32</b>	<b>96.06</b>	100	100	100	100	100	100	100	100	100	100	<b>100</b>	<b>100</b>
Infectious disease	5	95.65	95.01	<b>96.01</b>	<b>95.74</b>	95.43	95.22	85.43	85.43	100	100	100	100	100	94.04	<b>100</b>	<b>100</b>	100	100
Anatomical disease	13	69.18	77.22	70.15	70.20	<b>70.20</b>	<b>76.98</b>	79.06	79.06	87.60	87.60	95.86	89.53	90	80.19	90	95.24	<b>90.00</b>	<b>96.34</b>
Infectious + anatomical diseases	17	71.07	84.75	<b>73.13</b>	84.03	72.61	<b>84.98</b>	71.59	71.59	93.96	93.96	96.10	96.10	72.19	70	<b>90</b>	95.08	80	<b>97.03</b>

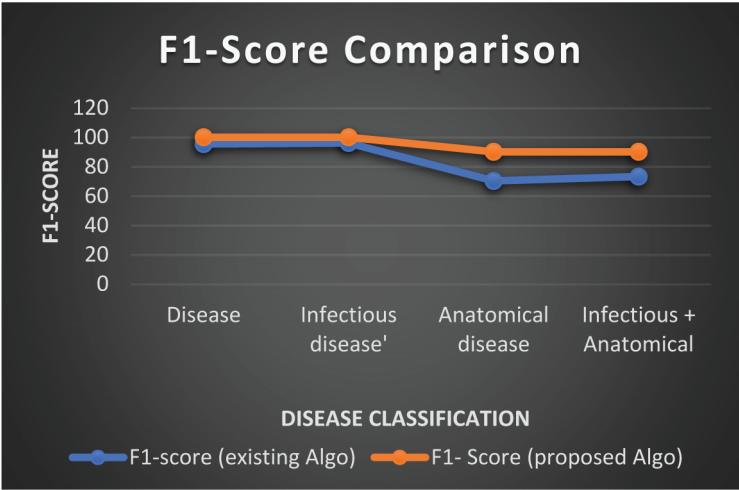
and Anatomical diseases, which consists of 17 classes. In this case, the highest accuracy and F1-score are 84.98 % and 73.13 % respectively and we have achieved the same as 97.03 % and 90% respectively. The comparison graph for the F1-score and AUC is as shown in Figures 6(a) and 6(b) respectively. We are comparing the metrics like F1-score and AUC disease wise not considering hidden layers.

The values in bold represent the highest F1 score and AUC for different experiments.

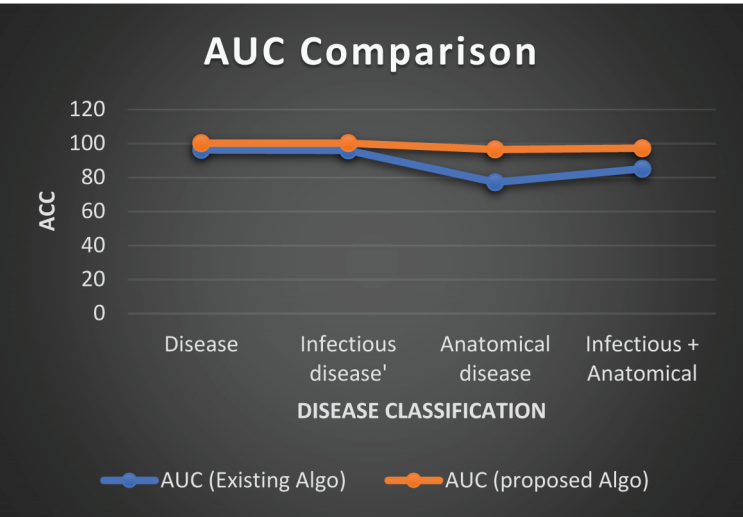
### 5.2.1 Discussion

The proposed algorithm gives better accuracy as compared to the existing algorithm. In all four experiments, the results are better. When the classes are less, the difference in accuracy and F1-score of the existing and proposed approaches is less. As the number of classes is increasing, the difference has improved. When the number of classes is 2, the difference is only 3-4% as shown in Table 2.

Figure 6. (a) Comparison of F1-Score (b) Comparison of Accuracy



(a)



(b)

Similarly, in the case of the 5 classes, the difference is the same. When the classes are 13, the difference is around 20%, and similarly in the last case also. The F1-score and Accuracy are better in all the disease classifications as compared to the existing one. This neural network trains on a reduced set of classes after ontology matching and thus upgrades the training process.

## 6. THREATS OF VALIDITY

There must be a certain degree of validity in the results presented in this research before it can be considered as a contribution to the area.

One threat of validity is the usage of Python libraries in this research. The library that implements the algorithms, sci-kit-learn, is what we rely on. The open-source library's simplicity is tempting, but it also poses a threat to validity because we have no control over its implementation. Another threat against validity in this research is how the data gets split. The splitting of data must be done in a balanced way if the results are to be trusted. An unreliable result may be obtained if there is an imbalance between distinct classes or labels. This can be avoided by balancing the amount of data for each category in a way that makes sense. One more threat is the Reliability of treatment implementation, which means that there is a possibility of varying implementation across researchers in their application of the treatment or between different time periods. As a result, it's necessary to use the same implementation, or as identical as possible, for different treatments or at different times. Results can also be affected by changes in the dataset.

## **7. CONCLUSION**

The result of this study shows that text classification with ontology gives a better result as compared to text classification without ontology using an Artificial Neural Network. A difference of 0% to 12% is observed in all the parameters, i.e. accuracy and F1 score with Ontology-based classification. When the classes are very less, the difference is less. As the number of classes is increasing the difference is improved. We have also compared our work with one of the research papers and we have achieved better results in terms of F1 score and accuracy for different classes for different hidden layers. In all four experiments, it has been observed that the accuracy and F1 score are better as compared to the results given by Althubaiti et al 2020. Disease ontology is used to categorize the symptoms of different diseases and to form relationships among them. The obtained text is used for ontology matching, and then finally for classification. This neural network trains on a reduced set of classes after ontology matching and thus upgrades the training process. Thus, our initial assumption proves to be valid. This research proved out to be beneficial however it won't be efficient enough in the case of a vast dataset where a brief medical history of the patients is provided as input. Therefore, there is scope to improve the neural network to function readily on similar vast datasets.

## **CONFLICT OF INTEREST**

On behalf of all authors, the corresponding author states that there is no conflict of interest.

## REFERENCES

- Abrori, M. (2016). Model Matematika Minuman Keras. *Jurnal Fourier*, 5(1), 1-9. <https://github.com/nltk/nltk>
- Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. J. (2011, June). Sentiment analysis of Twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)* (pp. 30-38). Academic Press.
- Althubaiti, S., Kafkas, Ş., Abdelhakim, M., & Hoehndorf, R. (2020). Combining lexical and context features for automatic ontology extension. *Journal of Biomedical Semantics*, 11(1), 1–13.
- Altinel, B., & Ganiz, M. C. (2018). Semantic text classification: A survey of past and recent advances. *Information Processing & Management*, 54(6), 1129–1153.
- Arbelaez, M. C., Versaci, F., Vestri, G., Barboni, P., & Savini, G. (2012). Use of a support vector machine for keratoconus and subclinical keratoconus detection by topographic and tomographic data. *Ophthalmology*, 119(11), 2231–2238.
- Bengio, Y., & LeCun, Y. (2007). Scaling learning algorithms towards AI. *Large-Scale Kernel Machines*, 34(5), 1-41.
- Bodner, R. C., & Song, F. (1996, May). Knowledge-based approaches to query expansion in information retrieval. In *Conference of the Canadian Society for Computational Studies of Intelligence* (pp. 146–158). Springer.
- Columbia. (n.d.). <http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html>
- Dagan, I., Karov, Y., & Roth, D. (1997). *Mistake-driven learning in text categorization*. arXiv preprint [cmp-lg/9706006](https://arxiv.org/abs/1907.06006).
- Dargan, S., Kumar, M., Ayyagari, M. R., & Kumar, G. (2020). A survey of deep learning and its applications: A new paradigm to machine learning. *Archives of Computational Methods in Engineering*, 27(4), 1071–1092.
- Dave, K., Lawrence, S., & Pennock, D. M. (2003, May). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web* (pp. 519-528). Academic Press.
- De Paz, J. F., Bajo, J., López, V. F., & Corchado, J. M. (2013). Biomedic Organizations: An intelligent dynamic architecture for KDD. *Information Sciences*, 224, 49–61.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6), 391–407.
- Fageeri, S. O., Ahmed, S. M. M., Almubarak, S. A., & Mu'azu, A. A. (2017, January). Eye refractive error classification using machine learning techniques. In *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)* (pp. 1-6). IEEE.
- Fellbaum. (1999). *WordNet*. Wiley Online Library.
- Garla, V. N., & Brandt, C. (2012). Ontology-guided feature engineering for clinical text classification. *Journal of Biomedical Informatics*, 45(5), 992–998.
- Green, S. J. (1999). Building hypertext links by computing semantic similarity. *IEEE Transactions on Knowledge and Data Engineering*, 11(5), 713–730.
- Jain, S. (2021). Understanding semantics-based decision support. CRC Press, Taylor& Francis Group.
- Kaur, P., & Sapra, R. (2013). Ontology based classification and clustering of research proposals and external research reviewers. *International Journal of Computers and Technology*, 5(1), 49–53.
- Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications*, 3(2), 85.
- Lee, Y. H., Tsao, W. J., & Chu, T. H. (2008, December). Use of ontology to support concept-based text categorization. In *Workshop on E-Business* (pp. 201-213). Springer.
- Lewis, D. D. (1998, April). Naive (Bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning* (pp. 4-15). Springer.

- Ma, J., Xu, W., Sun, Y. H., Turban, E., Wang, S., & Liu, O. (2012). An ontology-based text-mining method to cluster proposals for research project selection. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(3), 784-790.
- Malik, S., & Jain, S. (n.d.). Semantic Ontology-Based Approach to Enhance Text Classification. CEUR Workshop Proceedings.
- Ng, H. T., Goh, W. B., & Low, K. L. (1997, July). Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 67-73). ACM.
- Ruiz, M. E., & Srinivasan, P. (1997). Automatic text categorization using neural networks. *Advances in Classification Research Online*, 8(1), 58-68.
- Salton, G., & Yang, C. S. (1973). On the specification of term values in automatic indexing. *The Journal of Documentation*, 29(4), 11-21.
- Sanchez-Pi, N., Martí, L., & Garcia, A. C. B. (2014). Text classification techniques in oil industry applications. In *International Joint Conference SOCO'13-CISIS'13-ICEUTE'13* (pp. 211-220). Springer.
- Sanchez-Pi, N., Martí, L., & Garcia, A. C. B. (2016). Improving ontology-based text classification: An occupational health and security application. *Journal of Applied Logic*, 17, 48-58.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1), 1-47.
- Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.
- Waudby, C. J., Berg, R. L., Linneman, J. G., Rasmussen, L. V., Peissig, P. L., Chen, L., & McCarty, C. A. (2011). Cataract research using electronic health records. *BMC Ophthalmology*, 11(1), 1-15.
- WHO. (2010). *International Classification of Diseases (ICD)*? Available online: [http://www.who.int/classifications/icd/ICD10Volume2\\_en\\_2010.pdf](http://www.who.int/classifications/icd/ICD10Volume2_en_2010.pdf)
- Woźniak, M., Grana, M., & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16, 3-17.
- Wu, C. H., & Tsai, C. H. (2009). Robust classification for spam filtering by back-propagation neural networks using behavior-based features. *Applied Intelligence*, 31(2), 107-121.
- Yang, X. Q., Sun, N., Zhang, Y., & Kong, D. R. (2008, December). General framework for text classification based on domain ontology. In *2008 Third International Workshop on Semantic Media Adaptation and Personalization* (pp. 147-152). IEEE.
- Yu, F., Zheng, D. Q., Zhao, T. J., Li, S., & Yu, H. (2006, August). Text classification based on a combination of ontology with statistical method. In *2006 International Conference on Machine Learning and Cybernetics* (pp. 1042-1047). IEEE.