Analysis of Biomedical Entities extraction and classification using TensorFlow based on Neural Network

G. Suganya
Ph. D Research Scholar
Department of Computer Science
Bharathiar University
Coimbatore, India

Dr. R. Porkodi
Associate Professor
Department of Computer Science
Bharathiar University
Coimbatore, India

Abstract

Deep Learning is an important research and nowadays growing area. Deep learning does not require any process tasks or it require simply feature generation process instead; techniques are used for automatically extracted the features. The major biological applications are included in many research areas like biomarker development, drug discovery and prediction, gene annotation, medical image recognition, repurposing and structural biology, chemistry. Over the past decades biomedical entities classification have been focused by many research areas for improving the entities identification and classification accuracy, but still the problem persists due to the nomenclature and the new entities are identified day by day. The paper concentrated on the biomedical entities classification. There are different ways of entities identification approaches based on the entities. The methodology includes four phases: the first phase data collection and the second phase text data preprocessing; text are preprocessed using various text preprocessing library like NLTK; the processed data are classified using TensorFlow in neural network. An experimental study was done on various text based sentences which are collected from NCBI database. The overall performance of the proposed method is examined in terms of accuracy. The method exhibited 87% extracting entities classification which shows the significant performance.

Keywords: Deep learning, TensorFlow, NLTK libraries, Biomedical entities

I. Introduction

Deep is a technical term and it refers to the number of layers in Neural Network. Deep Learning is a subset of machine learning and it is a field of study which is dedicated to development of machines which would learn based on given inputs and achieving Artificial Intelligence inspired by human brain. The figure 1 shows the pictorial representation of Deep Learning. Deep Learning has found in many applications like computer vision, Natural Language Processing and Automatic Speech Recognition, etc.. The deep learning would mainly use both supervised and unsupervised learning with the combination of parametric and non-parametric model. Artificial Neural Network (ANN) or Neural Network (NN) consists of Multilayer Perceptron's (MLP) which contain one or more hidden layers with multiple hidden units. MLP is represented easily through Neural Network models using computation graphs called Direct Acyclic Graphs (DAG) [1].

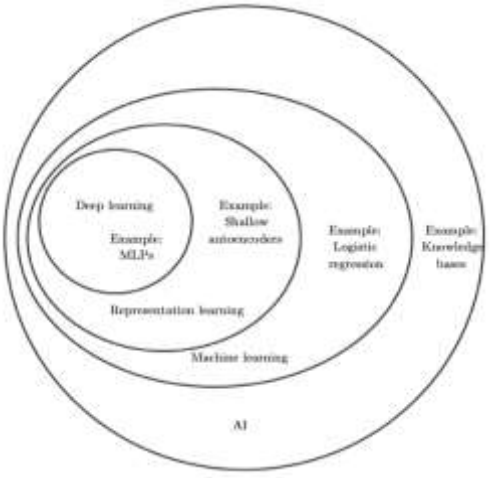

Figure 1. Diagram for deep learning

II. Deep Learning Methods

There are different deep learning architectures are present such as Convolutional Neural Network, Recurrent Neural Network, Autoencoders, Restricted Boltzmann Machine based Neural Network and Sparse Coding. The figure 2 shows the architecture of the deep learning methods. Text classification in Deep learning, Recurrent Neural Network, Recursive Neural network and Convolution Neural Network architecture are preferred [2]. The detailed description about each architecture are described in below section.

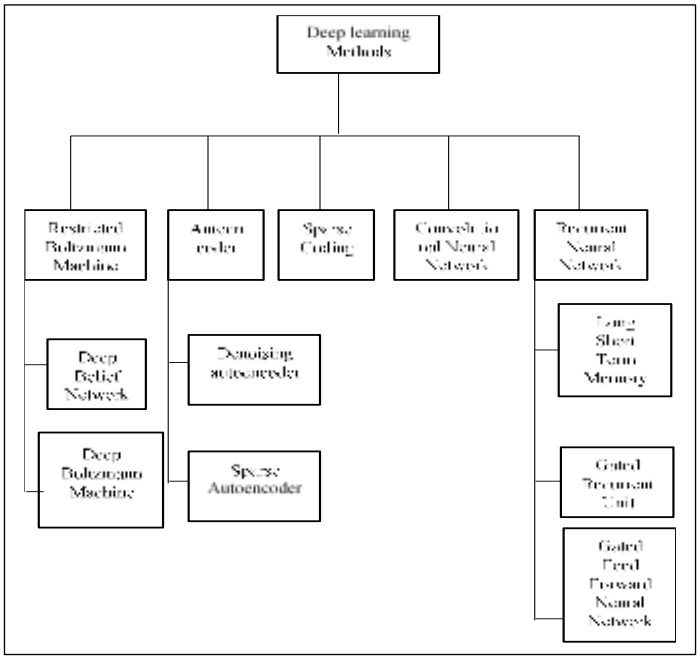

Figure 2. Deep learning methods

2.1 Restricted Boltzmann Machine (RBM)

The Boltzmann Machine (BM) is a form of log-linear Markov Random Field for which the energy function is linear. If the model contains more hidden units, can increase the capacity of Boltzmann machine. It restricts Boltzmann machines to those have without visible-visible and hidden-hidden connections [3]. The figure 3 shows the general view of Restricted Boltzmann machine network with visible and hidden units. There are two sub techniques are comes under in RBM are discussed below.

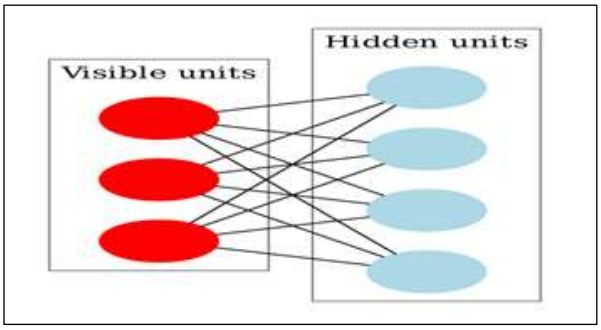

Figure 3. Restricted Boltzmann Machine

Deep Belief Network (DBN)

The Deep Belief Network is an Unsupervised probabilistic learning model. The model is composed of multiple layer of stochastic latent variables which is called as hidden units. It is a generative hybrid graphical model which is shown in figure 4. The model is pretrained with Greedy learning algorithm [4].

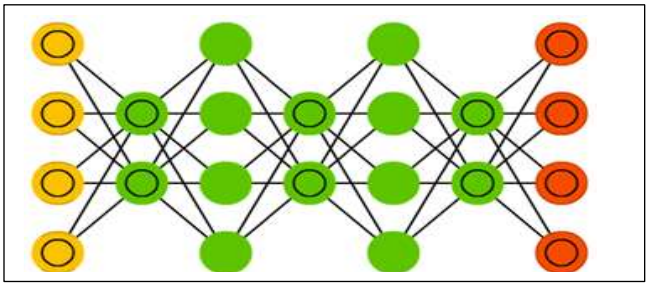

Figure 4. Deep Belief Network

Deep Boltzmann Machine (DBM)

The Deep Boltzmann Machine is an unsupervised, probabilistic, generative model with undirected connections between various layers. The graphical representation of DBM network which contains the visible units and multiple layers of hidden units is shown in figure 5. The connections are present only between units of the neighboring layers. DBM is a bipartite graph with odd layers on the side and even layers on the side. After learning the binary features in each layer, the model is fine-tuned by back propagation [5].

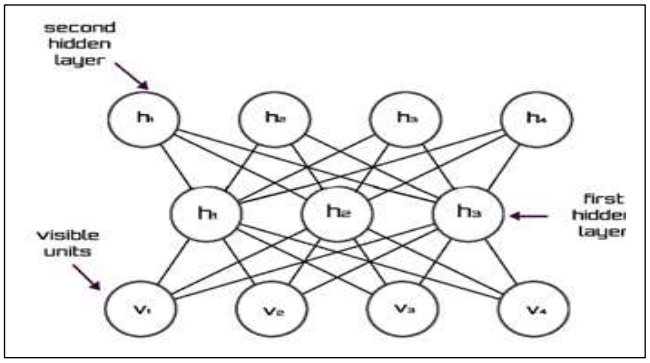

Figure 5. Deep Boltzmann machine

2.2 Autoencoder

An autoencoder is a neural network, is trained to attempt for copy the input as well as output. The model has a hidden layer which describes a code used to represent the input. The network consists of two parts such as i) encoder function ii) decoder function are shown in figure 6. The idea of autoencoders is the part of historical landscape of neural network. The method was used for feature learning or dimensionality reduction. This is the special case of feedforward network and trained with all same

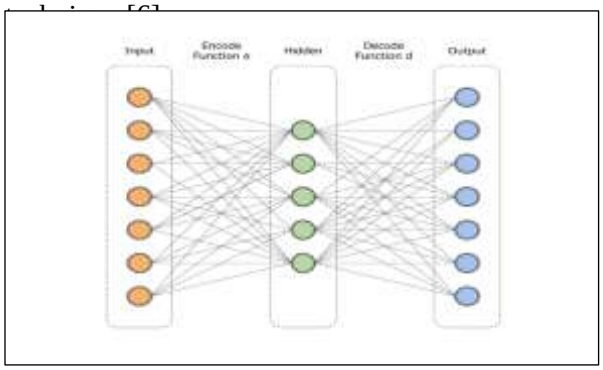

Figure 6. Autoencoder

Denoising Autoencoder

Adding penalty to the cost function, obtain autoencoder that should learn something useful by changing the reconstruction error in term of cost function [6]. The graphical representation of Denoising autoencoder is shown in figure 7.

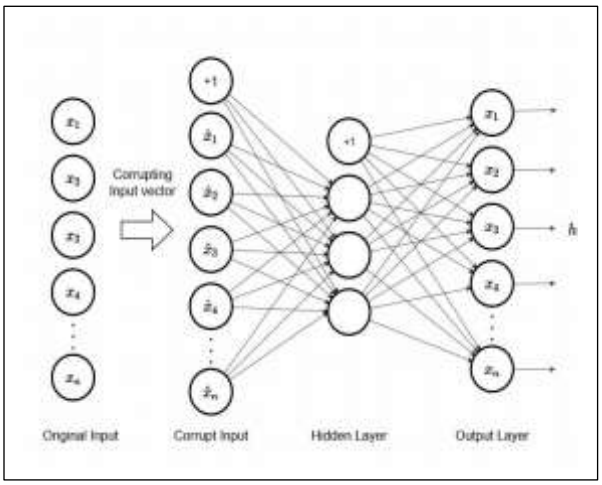

Figure 7. Denoising autoencoder

Sparse autoencoder

The method is used for learning the features of the classification. It is autoencoder whose training norm includes sparsity penalty on the code layer addition to the reconstruction error. The autoencoder has regularized to sparse must respond to unique features of the dataset it has been trained on, or act as identify function. Training is performed by copying task with sparsity penalty which can yield as useful features [6]. The figure 8 shows the sparse autoencoder network.

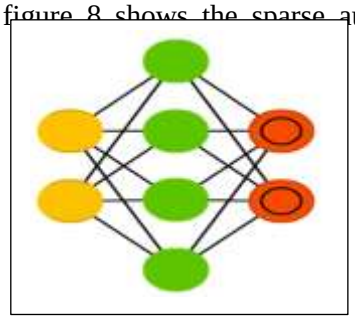

Figure 8. Sparse autoencoder

2.3 Sparse coding

The Sparse coding is an unsupervised method and study of algorithms which learn useful representation of any given data. This is the basic task in many research fields like signal processing, neuroscience and machine learning where it is used to learn a basis that enables a

sparse representation of given set of data, if one data exists [7]. The figure 9 shows the sparse graph from dense graph.

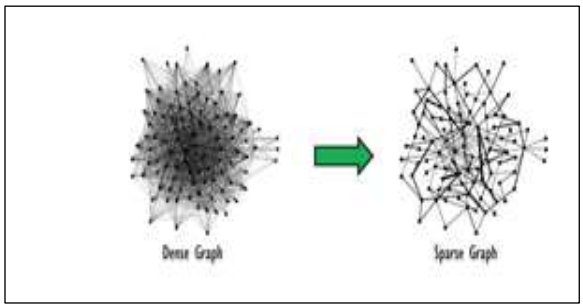

Figure 9. Sparse Coding

2.4 Convolutional Neural Network

It is used for processing of sequences of text and also used in image recognition system. The model is based on neural network system which represents the function of each features applied to words to extract high level of features. After extracting the features, this can be used in many applications like sentiment analysis, question answering and machine translation, etc. the figure 10 shows the block diagram of CNN model [8].

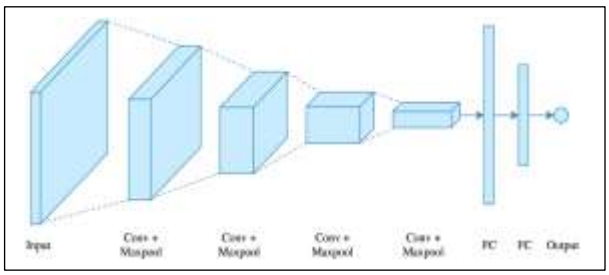

Figure 10. Convolution Neural Network

2.5 Recurrent neural network

In this approach the data can flow from in any direction and used to process sequences of inputs and possess internal memory because the proposed or produced model output is depending on present and recent scenario. Since the network finds the application which is in text analysis, DNA sequencing, speech recognition system, image related applications and language modeling, etc.[9]. The figure 11 shows the recurrent neural network model used for processing of text data.

The figure 12 shows the comparison between RNN with LSTM and GRU.

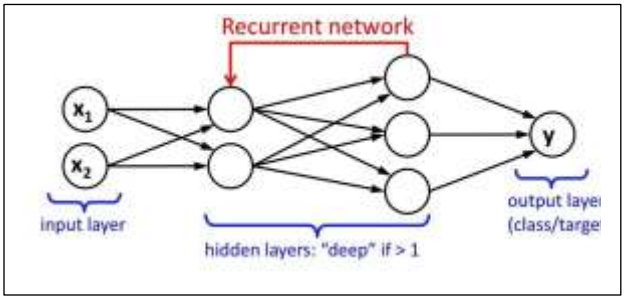

Figure 11. Recurrent Neural network

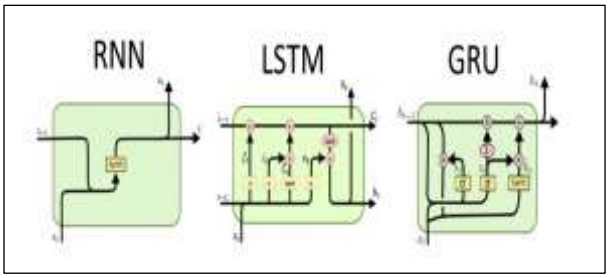

Figure 12. Comparison of RNN with LSTM and GRU

Long Short Term Memory (LSTM)

The method was introduced by Hochreier & Schmidhuber and it is capable of learning long-term dependencies. The system remembers the information for long period of time i.e remember the values over arbitrary time interval and it have chain like structure but the repeating module has different type of structure [10]. The common LSTM consists of cell, input gate, output gate and forget gate are shown in figure 13. These three gates regulate the flow of information. This is more suitable in classifying, predictions, etc. The system mainly developed for deal with the gradient vanishing problem [11].

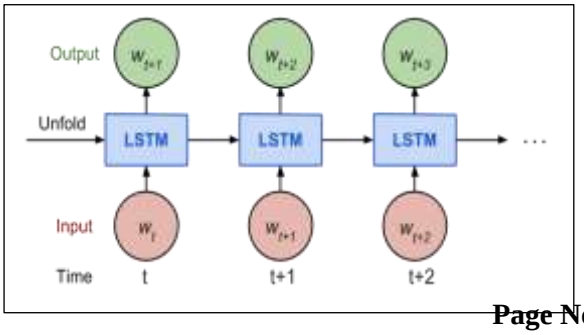

Figure 13. LSTM

Gated Recurrent Unit (GRU)

GRU is used for solve the vanishing gradient problem of a standard RNN, GRU uses update and reset gate. These two vectors decides what kind of information to be passed to the output gate. The advantages in this is that the system is trained to keep the information from long without washing or remove information which are not relevant for prediction [12]. The working process of GRU is depicted in figure 14.

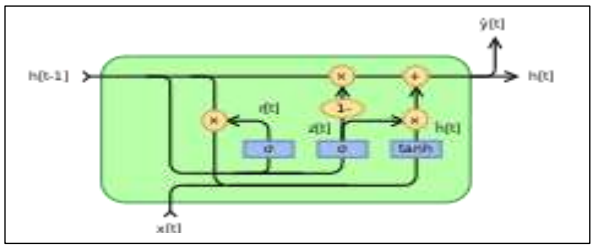

Figure 14. Gated Recurrent Unit

Feedforward Neural Network (FNN)

Feedforward Neural Network is an artificial neural network where the connection between the nodes which do not form a cycle in the network. This the first and simplest type of artificial neural network model. In this network, the information transfers only in one direction from the input nodes, through hidden nodes and output nodes [13]. There are no feedback connections which output of the model are fed back into itself [14]. The network often called as Deep feedforward network or multilayer perceptrons (MLP) or Multi-layered Network of Neurons (MLN). The general working principle of FNN is shown in figure 15 and the network is compared with RNN is shown in figure 16.

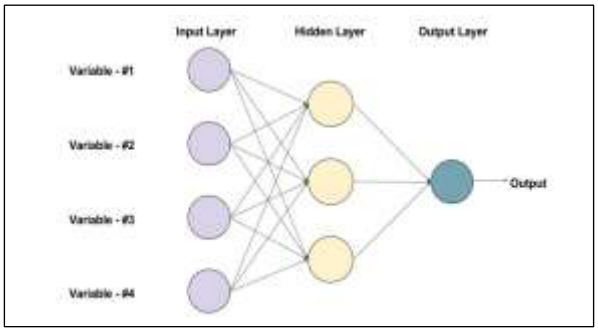

Figure 15. Feedforward Neural Network

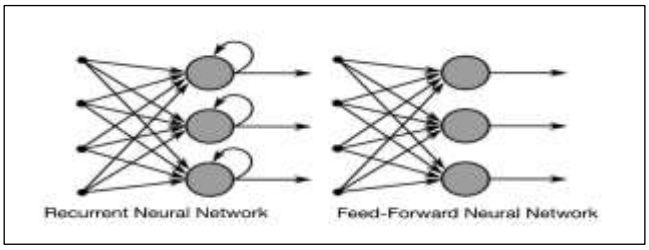

Figure 16. Difference between RNN and FNN

III. Methodology framework

The main objective of this is used to identify and extract the biomedical entities using neural network architecture. The methodology framework is given in figure 17. The framework consists of four phases: Phase 1 is data creation, Phase 2 is Preprocessing and convert into the model, Phase 3 is designing of neural network model, Phase 4 is testing the model.

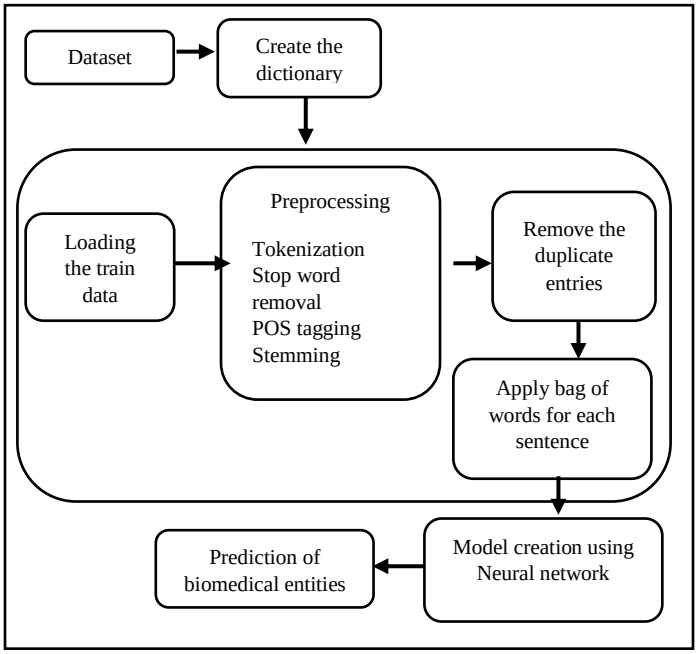

Figure 17. Framework of entities classification using neural network

Phase 1: Dataset creation

The dataset is collected from NCBI database for classifying the data into four categories like disease, drug, mutation and gene. The collected data format is represented in #1.

#1

```
{"content":"Several molecular effects of ketamine are linked to, and possibly underlie, its antidepressant actions: Activation of the mTOR pathway (Li et al., 2010).","entities":"Gene"}
{"content":"This is noteworthy since mTOR is a central controller of protein synthesis required for new synaptic connections (Gong et al., 2006).","entities":"Gene"}
{"content":"Phosphorylation of eukaryotic elongation factor 2 is inhibited, which augments subsequent expression of BDNF (Gideons et al., 2014).","entities":"Gene"}
{"content":"Phosphorylation of eukaryotic elongation factor 2 is inhibited, which augments subsequent expression of BDNF (Gideons et al., 2014).","entities":"Gene"}
{"content":"Activating mTOR, inhibiting Eef2 phosphorylation, and stimulating BDNF release should all stimulate synaptic plasticity and new memory formation within the brain.", "entities": "Disease"}
```

Phase 2: Text preprocessing

The needed NLTK libraries are imported for text preprocessing tasks like stemming, stopword removal, tokenization and bag words.

Stemming: Stemming is a process used for identifying the stem words which used for classify the sentences. The Lancaster stemmer is used to do perform stemming operation and the code is given in #2.

#2

```
stemmer = LancasterStemmer()
```

Tokenization: The punctuation is also removed from the sentences. It is used to split the sentences into the tokens using NLTK libraries as given in #3.

#3

```
words = []
classes = []
documents = []
ignore_words = ['?']
# loop through each sentence in our training data
for pattern in training_data:
    # tokenize each word in the sentence
    w = nltk.word_tokenize(pattern['content'])
    # add to our words list
    words.extend(w)
    # add to documents in our corpus
    documents.append((w, pattern['entities']))
    # add to our classes list
    if pattern['entities'] not in classes:
        classes.append(pattern['entities'])
```

Remove duplicates: Duplication of text data is removed and the same is given in #4.

#4

```
words = list(set(words))
classes = list(set(classes))
```

Bag of Words for the sentences: The Bag of Words model is also referred as BoW, It is used for extracting features from text which is for modeling like machine learning algorithms. The model is very simple, flexible and used in many ways for extracting features from documents. It describes the occurrence of words within a document. The complexity come both in simple and complex in deciding how to design the vocabulary of known words and how to score the presence of known words is shown in #5.

#5

```
bow(sentence, words, show_details=False):
    tokenize the pattern
    # bag of words
    bag = [0]*len(words)
    for s in sentence_words:
        for i,w in enumerate(words):
            if w == s:
                bag[i] = 1
                if show_details:
                    print ("found in bag: %s" % w)
    return(np.array(bag))
```

Phase 3: Neural network model- Biomedical entity classification

The neural network model is created with input nodes, 20 hidden layers and four output layers is represented in figure 18 and table 1.

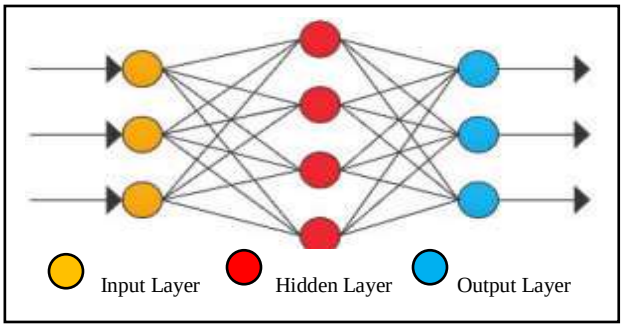

Figure 18. Neural Network

Table 1. Neural network details

Neural network model	Number of nodes
Hidden layer	20
Output layer	4

The neural network architecture is created with 20 hidden layers and 4 output layers. the model uses the softmax activation function with 10000 epochs and the dropout percentage is 0.2. The creation of training data is shown in #6.

#6

```
X = np.array(sentences)
y = np.array(classes)
train(X, y, hidden_neurons=20, alpha=0.1,
epochs=100000, dropout=False, dropout_percent=0.2)
```

Phase 4: Validating the model

Performance measure: The model is verified using the precision, recall and F-Measure. The formula for the corresponding measures is represented in equation 1, 2 and 3 respectively.

$$Precision = True\ Positive / True\ Positive + False\ Positive \quad \text{--Equation 1}$$

$$Recall = True\ Postive / True\ Positive + False\ Negative \quad \text{--Equation 2}$$

$$F\ Measure = 2 * Precision * Recall / Precision + Recall \quad \text{--Equation 3}$$

IV. Results and discussion

The neural network model is used to classify the biomedical entities using TensorFlow. The created model is tested with various epochs and dropout value. This section discusses the performance of the created architecture and is tested with test dataset with 50 sentences used for identifying the entities is shown in figure 19.

4.1 Prediction: The architecture was tested with the test data. The model prediction also verified and validated. The figure 20 shows the prediction of different entities or classes using neural network.

```
4 classes ['Drug', 'Disease', 'Gene', 'Mutation']
```

Figure 20. Different classes

The following words (Figure 21) are unique stemming words after applying Lancaster stemmer operation. The figure 22 shows the Bag of Words (BoW) result for the given corresponding sentence. The BoW array is found from unique stemming words which are trained form training data.

```
classify("Ketamine: A Neglected Therapy for Alzheimer Disease")
classify("One of the biggest breakthroughs in the study of depression has been the observation that ketamine, a NMDA antagonist drug used as an anesthetic and recreational drug, can reverse symptoms of depression within hours to days when given by intravenous infusion")
classify("Ketamine has significant neurological and psychiatric side-effects, particularly when given chronically or at high doses, including dizziness, drowsiness, confusion, feelings of dissociation, insomnia, hallucinations, psychosis, and impaired learning and memory.")
classify("No wonder that a search of Clinicaltrials.gov (carried out in April 2019) shows 152 trials of ketamine in depression, and 38 trials that involve various antidepressants in Alzheimer patients, but no registered trials that involve giving ketamine to Alzheimer patients")
classify("On the other hand, memantine is one of the most prescribed and most investigated drugs employed in Alzheimer patients-and both ketamine and memantine are non-competitive NMDA receptor antagonists with similar, though non-identical, pharmacologic profiles")
```

Figure 19. Test dataset

```
['memantine', 'Nikiforuk', 'Gideons', 'chronically', 'brain', ',', 'relief', 'augments', 'inhibiting', 'exhibits', 'block', 'intravenous', 'synthesis', 'factor', 'ability', 'restraint', 'ketamine', 'stimulate', 'kinetics', 'voltage-dependence', 'Alzheimer', 'can', 'anesthetic', 'eukaryotic', 'Eef2', 'feelings', 'Disease', 'Gong', 'expression', 'biggest', 'affinity', 'release', 'impaired', 'et', 'significant', 'confusion', 'during', 'in', 'stress', 'drowsiness', 'Neglected', 'recreational', 'Phosphorylation', '.', 'actions', '10', 'psychiatric', 'or', 'flexibility', 'molecular', 'linked', 'of', 'within', 'reverse', 'fast', 'the', 'extra-dimensional', 'ED', 'possibly', 'cognitive', 'doses', 'activity', 'psychosis', 'connections', 'depression', 'been', 'effects', 'central', 'required', 'formation', 'study', 'side-effects', 'when', 'Ketamine', 'Zanos', 'infusion', 'underlie', 'rats', 'repeated', 'Compared', 'neurological', 'This', 'are', 'channel-blocking', 'another', 'BDNF', 'mTOR', 'noteworthy', 'hours', '(', 'set-shifting', 'inhibited', 'from', 'breakthroughs', 'memory', 'since', 'NMDA', 'Popik', 'synaptic', 'which', 'drug', 'In', 'Gould', 'Therapy', 'Activating', 'stimulating', 'hallucinations', 'Several', 'closed', 'mg/kg', '2', 'antagonist', 'One', 'symptoms', 'rapid', 'all', 'egress', 'elongation', 'preserved', 'allowing', 'dissociation', 'observation', 'to', 'plasticity', 'permitting', 'an', ':', 'and', 'antidepressant', 'well', 'high', 'dizziness', 'trapping', 'subjected', 'protein', 'Activation', 'strong', 'that', 'A', 'lower', 'channels', 'phosphorylation', 'learning', 'subsequent', 'controller', 'pathway', 'particularly', 'insomnia']
```

Figure 21. Stemming Words

The figure 23 shows the predicted output of neural network for the test data. The prediction of the network model is varied based on the

The created neural network based on tensorflow model tested with 50 test dataset. This model achieved 86.5% for classifying biomedical entities.

Sentence: Several molecular effects of ketamine are linked to, and possibly underlie, its antidepressant actions: Activation of the mTOR pathway.
BoW:
array ([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0])

Figure 22. BoW Representation

Sentence: No wonder that a search of Clinicaltrials.gov (carried out in April 2019) shows 152 trials of ketamine in depression, and 38 trials that involve various antidepressants in Alzheimer patients, but no registered trials that involve giving ketamine to Alzheimer patients:
classification: ['Drug']
Sentence: On the other hand, memantine is one of the most prescribed and most investigated drugs employed in Alzheimer patients-and both ketamine and memantine are non-competitive NMDA receptor antagonists with similar, though non-identical, pharmacologic profiles.
classification: ['Drug']
Sentence: Phosphorylation of eukaryotic elongation factor 2 is inhibited, which augments subsequent expression of BDNF
classification: ['Gene']

Figure 23. Predicted output

Each entities precision, recall and f-measure are shown in Table 2. From 50 test data, Gene, Disease, Drug and Mutation category has totally 20, 9, 13 and 8 sentences are present respectively. Gene achieves the 70% of precision, 87% of recall, 78% of f-measure and 78% of accuracy. Similarly, Disease achieves the 100% of accuracy for Drug and Mutation achieves 84%, 100% of precision and 73%, 80% of recall and 78%, 89% of f-measure respectively. Overall, the system achieves 87% of accuracy. The same is depicted in figure 23.

dropout value and alpha rate. The network is trained with 1000 biomedical entities sentences.

Table 2 Evaluation of Domain entities Specific result

Measures	Precision	Recall	F-Measure	Accuracy
Gene	0.7	0.87	0.78	0.78
Disease	1	1	1.00	1.00
Drug	0.84	0.73	0.78	0.78
Mutation	1	0.8	0.89	0.90
Overall	0.89	0.85	0.86	**0.87**

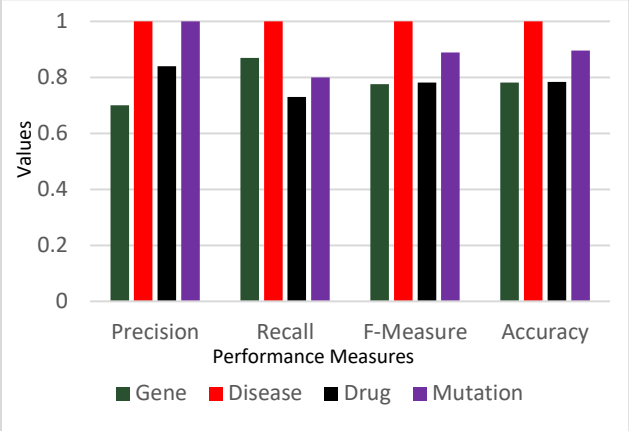

Figure 23. Performance measures

V. Conclusion

Deep learning is a subfield of machine learning algorithms with the root of Artificial Neural Network. It is one of the best, mostly used and successive algorithms. The objective of the work is to study the deep learning method for classifying the biomedical entities. the methodology consists of four phases such as data modelling or creation, text preprocessing, designing of neural network and validation.

The method used the TensorFlow library for the identification. the model identifies and classifies the biomedical entities into various

categories like gene, drug, mutation and disease. The created model is evaluated with the various performance measure like precision, recall and f-measure. The model is tested with the 50 sentences and validated with the different values for epochs, alpha and dropout percentage. The epochs are ranged from 10000 to 20000; alpha rate is from 0.1 to 0.5; dropout percentage is ranged from 0.2 to 0.5. overall the model achieved 87% of accuracy. In future, the model is extended to identify the hidden relationships among the biomedical entities.

VI. References

[1] R.A. Sahner, K.S. Trivedi, Performance and Reliability Analysis Using Directed Acyclic Graphs, IEEE Transactions on Software Engineering, SE13(10), 2002.

[2] Kotsiantis, Ikonomakis, et.al., "Text Classification Using Machine Learning Techniques", Volume 4, August 2005.

[3] Restricted Boltzmann Machines (RBM)- Deep learning 0.1 documentation

[4] Renu Khandelwal, "Deep learning - Deep Belief network (DBN)", 2018.

[5] Renu Khandelwal, "Deep learning - Deep Boltzmann Machine (DBM)", 2018.

[6] https://www.deeplearningbook.org/contents/autoencoders.html

[7] Sanjeev Arora, Rong Ge, Tengyu Ma, Ankur Moitra, "Simple, Efficient, and Neural Algorithms for Sparse Coding", arXiv:1503.00778v1 [cs.LG], 2015.

[8] Convolutional Neural Network Wikipedia

[9] Recurrent Neural Network Wikipedia

[10] Understanding LSTM networks, 2015

[11] Long short term memory Wikipedia

[12] Simeon Kostadinov, "Understanding GRU Networks", towards data science, 2017.

[13] Tushar Gupta, "Deep Learning: Feedforward neural network

[14] Feedforward Neural Network Wikipedia