International Journal of Advanced Science and Technology
Vol. 29, No. 7s, (2020), pp. 4661 - 4671

Semantic Relations Extraction in Biomedical Text Using Deep Learning Techniques

[1] **Ayush Eshan,**[2] **Srilekha Vinjamara,**[3] **Srinivasan R**

[1][2][3] Department of Computer Science and Engineering SRM Institute of Science and Technology Chennai, India

Abstract

Deep Learning is an emerging field of machine learning that has several use cases. Use of deep learning in the biomedical field is one such use case. Semantic relation- ship extraction is a natural language technique that seeks to identify the relationship between entities in a piece of text. The proposed research is aimed towards understanding the semantic relationship between diseases and treatment in a given biomedical sentence. The state-of-the-art model to achieve the goal has been done with the help of machine learning techniques, obtaining an accuracy of 90.72% for 3-class classification [14]. In this paper, an initiative has been taken to do the same with the help of deep learning approaches. This paper aims at identifying semantic relationships in biomedical text and various deep learning architectures using Rosario and Hearst dataset. The train accuracy obtained for a 9-class classification is 82.21%, whereas for a balanced binary classification is 98.24%.

Keywords: *Deep Learning, Text classification, Natural Lan- guage Processing (NLP), Recurrent Neural Network (RNN), Long-short term memory (LSTM), Hidden Markov Model (HMM)*

1. Introduction

Deep learning belongs to a broad category of machine learning techniques which aim at describing the correlations among data points. Deep Learning may be called hierarchical or highly organized learning. Various kinds of Deep Learning are controlled, semi-supervised or unsupervised.

Architectures of Deep Learning include Restricted Boltz- mann Machines, Artificial Neural Networks which are linear in nature and Recurrent Neural Networks like RNN and LSTM which are repetitive in nature and hence store information or obtain correlations among data points. Deep Learning has been made use of in many domains such as NLP, Named Entity Recognition, social network analysis, machine translation and real-life applications such as stock market prediction, speech analysis or cancer prediction. These research works seem to have demonstrated superior intelligence in some cases and has offered exceptional results.

Natural language processing is a computer science field concerned with interactions with the computer and human language[1]. Current trends indicate that deep learning is now used in tandem with natural language translation. Natural language processing problems that have been solved using machine learning models have often resulted in very shallow models trained on very high dimensional data and sparse features [2].

A wave of neural network processes are now able to deliver state of the art performance in many machine learning models. These neural networks have multiple interconnected layers which formed the basis of deep learning models [3].Earlier neural network language models were mainly shallow lan- guage neural network models [4][5].On the other hand, newer neural network models tend focus on language models with deep structure, like recursive neural network language models [6][7] and structured resultant layer neural net structures [8][9].

Many language-processing systems were developed in the early days by hand-coding a series of rules: for example, by writing grammars or by designing heuristic rules for stem- ming. Much natural language processing research has focused heavily on machine learning since the so-called 'statistical boom' in the late 1980s and mid-1990s. Alternatively, the machine-learning method calls for the use of mathematical inference to systematically learn these laws by evaluating big corpora (the plural form of corpus, is a collection of records, possibly with annotations from humans or computers) using common real world cases. These algorithms take a large set of features that are generated from the input data as their input. Most of the oldest-used implementations, such as decision trees, produced collections of harsh if-then standards that were identical to those of conventional scripted guidelines then. Nevertheless, there has been increased focus on works related to mathematical models which make simple, probabilistic decisions based on attaching real-evaluated biases and masses to every input component. These models have the advantage when such a model is used as a part of a broader process, then they will convey the relative confidence of several different alternative responses rather than just one, providing more accurate outcomes. The difference between the traditional NLP approaches and deep learning is that the former uses bag-of-words model, and the latter uses RNN word framework to prepare embedding terms [10]. This study used two styles of recurrent neural network structures: the Recurrent Neural Network (RNN) and the Long Short Term Memory (LSTM). The point to remember here is: LSTM is an RNN of a different type [12].

An RNN is an artificial neural net construction proposed for study of the time-data series in the 80's (Rumelhart, 1986; Werbos, 1988; Elman, 1990). The configuration of the network is similar to that of a standard multilayered perceptron, except that it makes contacts between hidden units connected with a time delay. The model would maintain awareness of the past through these connections, enabling it to recognise temporal correlations between events that are distant in the data [13]. LSTM is one of the most used, recurrent, Deep Learning neural network architecture. Schmidhuber invented it in 1997 [11], and by introducing three gated modules, this prevents the vanishing gradient problem: forget, input and output gates, which can effectively regulate the memory of its past states. Both RNNs and LSTMs have been used in this research to compare which model can provide better results in the extraction of semantic relationships in biomedical texts.

The work done in this paper aims to use Deep Learning approaches in a skewed dataset. The main goal is to achieve high accuracy with unbalanced class label distribution. Since the dataset is biomedical in nature, Natural Language Process- ing techniques like one-hot encoding, padding have been used along with Deep Learning approaches like RNN and LSTM. Different combinations have been used to determine the best method for identifiying semantic relations in biomedical cor- pora.

The state-of-the-art method and the dataset used for the same is mentioned in Section II of the paper. Following this section, there is proposed architecture in Section III which describes the dataset used in this paper along with the proposed dataset split. Also an outline of the entire architecture of this research work has been mentioned in this section. In the next section i.e. Section IV, multiple feature extraction techniques, which have been experimented on, have been discussed. Sec- tion V of the paper describes Deep Learning Techniques. All of the models have been implemented and the best model was found to have been using Recurrent Neural Network (RNN). Advantages of each of the architecture models was mentioned and a use case in which it is most suitable has been described. In Section VI, evaluation and results have been described with a detailed analysis of the particular model chosen. Finally, the last section of this paper talks about the conclusions drawn from the experiments performed in this paper and some ways in which this work can be extended in the future.

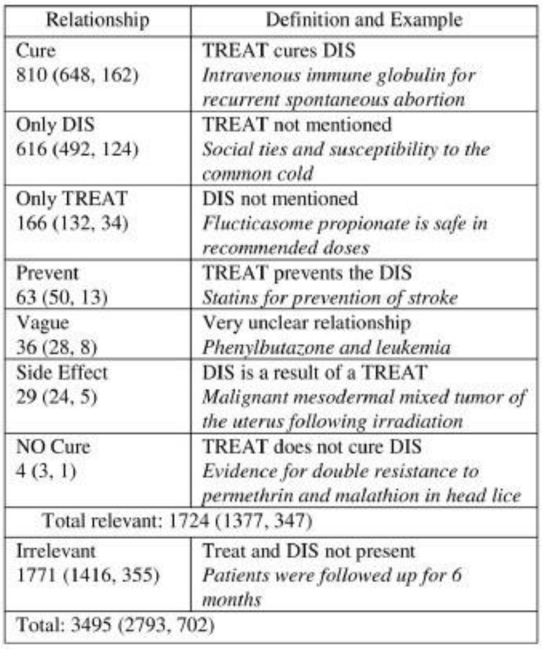
2. Literature Survey

2.1 Inference from the Survey (State of the Art

The most important recent research work is the work that Rosario and Hearst have done [14]. The data set comprises of phrases from abstracts of Medline which are annotated with disease and treatment entities and nine relational associations between diseases and treatments. Fig. 1 shows the elaborate split-up of the dataset.

The dissertation focuses mainly on semantic relationships for diseases and treatments. HMM and maximum entropy models have been used to perform both the entity identification and discrimination relationship tasks. Their techniques of representation are focused upon contextual words, part of speech knowledge, phrases, and a medical ontology — Mesh terminology. The tasks dealt with in this research are the extraction of information and the extraction of relation. In the scientific literature, molecular position (Craven), mutation- disorder interaction (Ray and Craven), and illnesses and medi- cations (Srinivasan and Rindflesch) was historically discussed, but with an emphasis on clinical activities.

The datasets used in particular biomedical tasks usually have short texts, mostly sentences. This is the case of the first two previously mentioned related works. The tasks also include recognition of the co-occurring relationships between persons in the same sentence.



Relationship	Definition and Example
Cure 810 (648, 162)	TREAT cures DIS *Intravenous immune globulin for recurrent spontaneous abortion*
Only DIS 616 (492, 124)	TREAT not mentioned *Social ties and susceptibility to the common cold*
Only TREAT 166 (132, 34)	DIS not mentioned *Flucticasome propionate is safe in recommended doses*
Prevent 63 (50, 13)	TREAT prevents the DIS *Statins for prevention of stroke*
Vague 36 (28, 8)	Very unclear relationship *Phenylbutazone and leukemia*
Side Effect 29 (24, 5)	DIS is a result of a TREAT *Malignant mesodermal mixed tumor of the uterus following irradiation*
NO Cure 4 (3, 1)	TREAT does not cure DIS *Evidence for double resistance to permethrin and malathion in head lice*
Total relevant: 1724 (1377, 347)	
Irrelevant 1771 (1416, 355)	Treat and DIS not present *Patients were followed up for 6 months*
Total: 3495 (2793, 702)	

Fig. 1. Inference of Dataset. This table gives an overview of the type of sentences in the dataset and also the category that each of the sentences belongs to.

3. Proposed Architecture

3.1 Dataset Description

The dataset being used in this research is the Rosario and Hearst dataset which contains 3655 sentences. These sentences can fall into any of the following 9 categories:

Cure means the sentence contains both the disease and the treatment for the disease and describes the treatment as the cure for the disease. Only DIS means the sentence only contains the disease as an entity. Only TREAT means the sentence contains only the treatment as an entity. Prevent means the sentence contains the disease and the prevention for it too. Vague means the relationship between the disease and the treatment is unclear. Side effect means the disease is caused as a side effect of some treatment. No cure means the disease mentioned in the sentence has no cure. Irrelevent means both disease and treatment are missing from the sentence. These categories give us an idea of the type of data being dealt with, in this paper. Along with it, the distribution of data across each of the categories(classes) becomes clear and evident for use, whether uniformly distributed or skewed.

Dataset split used in this paper (experiments):

- Train set: 0.64(0.8 in total, 0.2 of training set is used for validation)
- Test set: 0.2
- Validation set: 0.16

The number of sentences falling into each category varies and therefore the dataset is found to be highly skewed. For training the deep learning model, we have taken 2340 sentences. Another 731 sentences were taken for testing and

584 sentences were taken for the purpose of validation.

This split helped increase the accuracy numbers by a small range, than the other dataset split configurations where the performance was constant or there was a marginal downgrade.

3.2 Text Pre-processing

Pre-processing involves several tasks. Not all tasks are important for every dataset

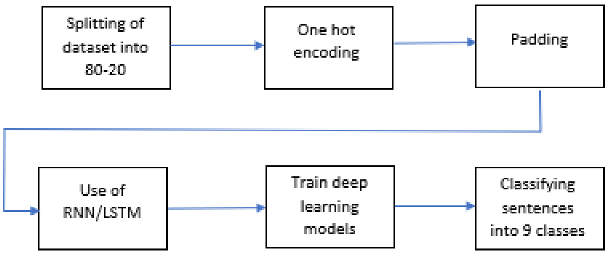

Splitting of dataset into 80-20
One hot encoding
Padding
Use of RNN/LSTM
Train deep learning models
Classifying sentences into 9 classes

Fig. 2. Proposed Architecture Pipeline

3.2 Workflow

These tasks are selected based on the type of data. However, when preprocessing is done over text data, some of these tasks tend to be standard procedures. Few steps involve splitting the text data into small tokens which may be words or phrases. These tokens are then passed on to another process wherein insignificant words like ‘a’, ‘this’, ‘that’ etc. are removed. This process is usually called stop word removal and the words like ‘a’, ‘this’, ‘that’ etc. are called stop words. Further, depending upon the data, a process called stemming might also be applied as a standard preprocessing method. In stemming, the words are converted to their root form. Fig. 2 gives an outline of the entire pipeline process followed in the proposed architecture in this research.

After preprocessing, the processed data is obtained that can now be used for classification. However, before proceeding with the classification process, changes to the structure or representation of the data is needed such that it can be suitably fed to the required classifier. This method is called indexing. When it comes to text classification, the words or phrases are used to represent a document. A word vector is generally used to represent a document. Few such common methods are one-hot encoding, Word2vec, GloVe model which represent a document using a word vector. The value or significance of the term is calculated as per the frequency of each of the terms in the records. This is followed by padding of all sentences to the length of the longest sentence in the corpus, to ensure balanced nature of the data and reduce sentence-level skewness.

After cleaning and preprocessing is done, data is then finally taken on to the classifier. Different types of classifiers are available for text classification. Some of the most important ones are Decision Tree Classifiers, Support Vector Machines, Neural Network Classifiers, Probabilistic Classifiers, etc. This paper proceeds to use Neural Networks as the classifier, since the dataset being used is large and the accuracy values obtained by this classifier are promisingly higher.

4. Feature Extraction

Word embedding is a type of interpretation of words that allows for a related representation of words with specific meaning. This is a structured text depictions which is expected to be one of the crucial advancements for the excellent growth of deep learning strategies that tackle NLP problems.

Some of the Word Embedding approaches are:

4.1 Embedding Layer (One-hot encoding)

The embedding layer is an embedding term which is learned on a particular natural language processing mechanism along- side a neural network system, such as vocabulary processing or document detection.

It needs cleaning and preparing document text in such a way that each word is one-hot encoded. One-hot encoding refers to dividing the column which contains numerical categorical data into several columns depending on the number of categories present in that column. Every column has a binary value, either
0 or 1, according to the column it has been placed in [15]. The vector space, such as dimensions 100 (used here), is defined as part of the model. The vectors are initialised with small random numbers. The embedding layer is put on the front of a neural network using the Backpropagation algorithm, and is constructed supervisely

4.2 Word2Vec

Although Word2vec is not a deep neural network, it does translate text into a numerical form deeply understood by the neural networks. Word2vec's intent and utility is to link vectors in a vector space together with the related terms. Which is, similitudes can be mathematically observed. Word2vec generates vectors that show numerical representations of word properties, such as the sense of the meaning of the individ- ual entities[16]. It does so without consulting anything else. Word2vec may make highly precise conjectures about the meaning of a word based on previous instances provided it has adequate proof, context, and meanings. These meanings may be used to equate a term with other words, or to identify records by category and cluster. Such assumptions can be used to match a word or cluster records with other words, and to identify them by subject. Word2vec's neural net output is a dictionary in which each object has a vector bound to it that can be loaded into or explicitly queried into a deep learning network for defining relationships among words.

4.3 GloVe

GloVe (Global Vectors) is a model where abstract words are interpreted. The model is an unsupervised training al- gorithm for tokens to construct vector representations. This is done by translating terms into a conceptual area where the difference between terms is linked to the similitude of semantics. Learning on consolidated universal expression- word co-occurrence stats is conducted on a corpus, and the resulting interpretations reveal fascinating linear vector space substructures[17]. In Stanford it's an open-source program. It incorporates characteristics of two model groups as the log- bilinear regression model for the learning of unsupervised representation of terms. The model families are universal vector factorization and strategies of windowing the specific context.

5. Deep Learning Techniques

The implementation uses one module based on Deep Learning algorithms like RNN-Recurrent Neural Network and LSTM-Long Short Term Memory.

5.1 Recurrent Neural Network (RNN)

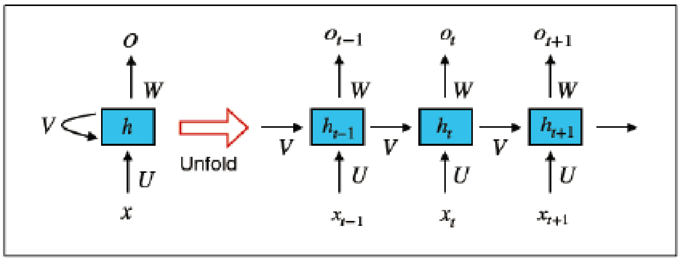


Fig. 3. RNN Model

It is the architecture of deep learning which is used for this research purpose. RNNs are advanced, neural-based ap- proaches that are efficient for processing sequential informa- tion. A Recurrent Neural Network applies a function recur- sively to each instance of an input sequence, based on the results of previous calculations. Usually, these quantities are expressed by a specified-size tokens map, which would be distributed chronologically to a recursive sequence. A basic structure for RNNs is shown in Fig. 3.
The key strength of an RNN is the ability to memorize

past computation outcomes and use the knowledge in current computation. This makes RNN models ideal for modeling contextual dependencies in arbitrary length inputs, in order to create a proper input composition. RNN is often used to evaluate numerous NLP operations, like machine translation, titling of frames, and simulation of language families. The input that an RNN expects is usually one-hot encoding or embedding of words.

RNN’s principal benefit over ANN is that RNN can model data sequence (i.e. time series) such that each observation can be considered to be based on previous ones. On the contrary, data series can not be modelled by ANN. ANN is useful only if each sample is believed to be independent of prior and subsequent ones. This is the reason for RNN to be one of the best fits for the model used in this paper with respect to biomedical sentence corpora.

5.2 RNN Variants (LSTM)

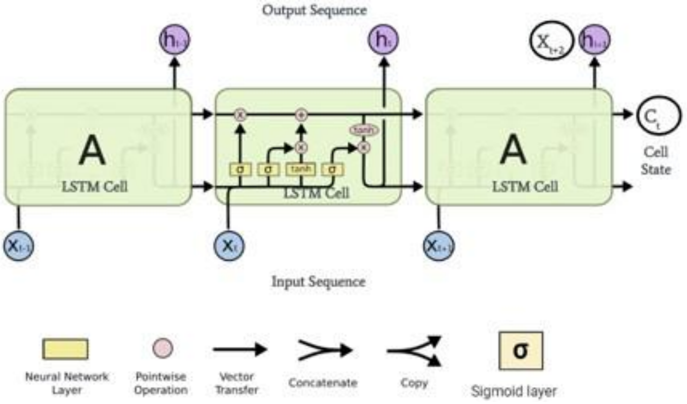


Fig. 4. LSTM Model

An LSTM as shown in Fig. 4, consists of three gates (input, forget, and output gates), and the hidden state is deter- mined by a combination of three. Different Long Short Term Memory based models for sequence mapping (via encoder- decoder frameworks) have been proposed which are suitable for machine translation, text summarization, human conversation modeling, answering questions, generating image-based language, among others.
One of the main advantages of LSTM is gap-length insensitivity. RNN and HMM (Hidden Markov Models) depend on the hidden state prior to emission/sequence condition. If we try to

forecast the series in place of 10 after 1,000 cycles; the algorithm has overlooked and forgotten the starting point by then. Theoretically, LSTM outperforms RNN in long sequences, but in the experiments performed it has been noticed that RNN outperforms LSTM when we use GloVe embeddings. The possible reason could be due to the skewed nature of the dataset wherein RNN remembers the weighted links and semantic information between almost all biomedical entities which might have been overlooked incase of LSTM due to gap-insensitivity.

5.3 Auto-Encoders

An auto-encoder which is an encoder-decoder network, belongs to a kind of Neural Networks. It is used autonomously to learn features. An auto-encoder's objective is to learn a specification (lossy compression) for a set of data, usually for dimension reduction [18][19][20], by enabling the network to not care about a signal termed as ”noise”. An augmentation side is taught along with the reduction side, where the auto- encoder tries to produce a representation as similar to its original input as possible. Recently the idea of auto-encoder has been more commonly used for Computational data pattern. In the 2010s, some of the most efficient AI involved minimal auto-encoders mounted within deep neural networks.

5.4 Early Stopping (added feature to enhance algorithm)

One challenge with developing neural networks is to pick the range of learning epochs to be used. Many epochs can result in the learning dataset being overfit, whereas too little can lead in the sample being underfit. Early stopping is a strategy that enables one to grant an infinitesimal amount of learning epochs and stop training until model output on a holdout validation dataset begins advancing. It is also called a type of mechanism for regularization, allowing the neural net to minimize overfitting.

6. Evaluation And Results

This Paper Has Focused On Understanding The Semantic Relationship Between Disease And Treatment In A Given Sen- Tence. The Curves In The Following Loss Curves Denote The ”Correctness” Of A Model And Depict How Well And Fast A Model Converges (As Depicted By The Number Of Iterations, Called Epochs).

Table I
Performance Using One-Hot Encoding

SL NO.	RNN vs. LSTM		
	Algorith	***Train***	***Test***
1	RNN	74.33	67.74
2	LSTM	74.80	66.09

Accounting for a skewed dataset, experiments have been performed on all the 9 classes using RNN and LSTM. This paper has also made use of word embedding techniques such as one-hot encoding in the embedding layer, Word2vec and GloVe (Global Vectors).

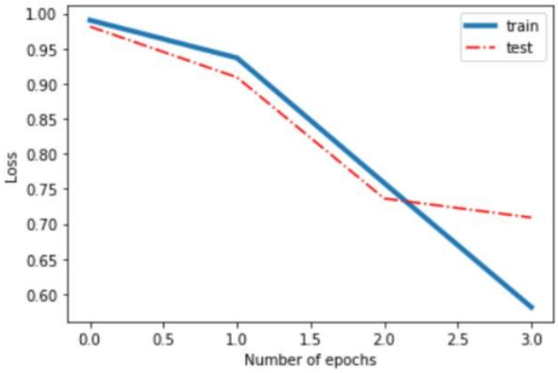


Fig. 5. Loss curve using One-hot encoding.

The focus of this research is to compare different Deep learning models and compare the pros and cons of using each model for the problem statement at hand. In the tables given below, the performance of each model has been measured using accuracy.

$$\text{Accuracy}= (T P + F P) / (T P + F P + T N + F N) \qquad \text{---- (1)}$$

For every category true positives(TP), true negatives(TN), false positives(FP) and false negatives(FN) are defined as de- picted in Equation (1). True positives is the count of sentences that were correctly classified into its category. False positives are the sentences that were classified wrongly into a category. True negatives is the count of sentences that were wrongly put in some other category by the deep learning model. False negatives is the number of sentences which are not put in the category that they belongs to.

Table II
Performance Using Word2vec

Sl No.	Rnn	Vs.	
	Algorith	*Train*	*Test*
1	Rnn	70.26	67.26
2	Lstm	76.72	67.12

TABLE III
Performance Using Glove

SL. NO.	RNN	vs.	
	Algorith	*Train*	*Test*
1	RNN	82.21	74.09
2	LSTM	80.29	72.26

The following observations are made:

• One-hot encoding (refer Fig. 5 and Table I): One hot encoding does not depend on any information as it assigns a unique integer to every word in the set of words in a document.

• Word2vec (refer Fig. 6 and Table II): Word2vec relies completely on local information of the language model. It generally consists of wikipedia vocabulary. This means that the semantics of any given word in a sentence is only going to be affected by the words in that sentence that surround the said word.

• Glove (refer Fig. 7 and Table III): Glove relies on word embeddings that is not dependent on only local informa- tion. Word embeddings can be of various dimensions like 50,100, etc. There word embeddings make use of local information, global information as well as term frequency inverse document frequency (tf-idf).

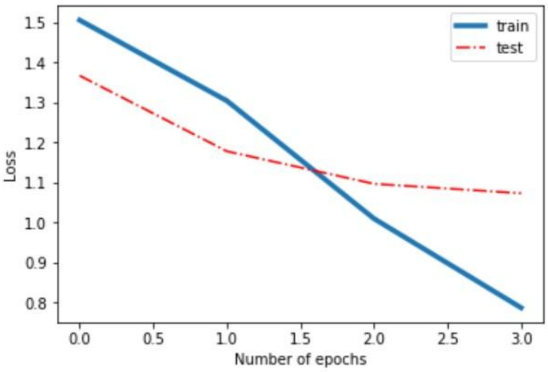


Fig. 6. Loss curve using Word2vec.

$$Cross\ entropy = -\sum_{c=1}^{M} y_{o,c} \log(P_{o,c}) \qquad (2)$$

where,
M - number of classes
log - the natural log
y - binary indicator for observation o
p - predicted observation o of class label c

The loss function used in this paper is Sparse Categorical Cross Entropy. The formula for the same has been shown in Equation (2).
The above results conclusively show that RNN performs better than LSTM for the dataset in consideration. Also, RNN out-performs auto-encoders in this case. Early stopping is used a means to not overfit the model and generalise on new test data given to the model. Another significant analysis that can be made is that Glove word embedding gave the best accuracy for the dataset used as a part of this research. A skewed dataset further prevents all the three preprocessing methods to yield optimal results.

7. Conclusion

Deep Learning has proved to be a significant and effective strategy for the problem at hand. It has helped to achieve a bet- ter understanding between biomedical entities and their relationships amongst each other. The understanding of semantics

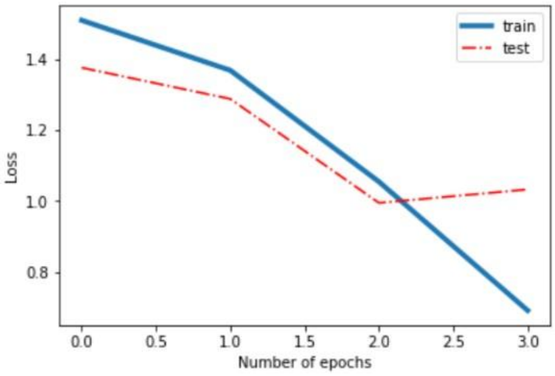


Fig. 7. Loss curve using Glove.

and sentence structure still seems to be a hurdle which needs to be tackled more efficiently. However with improvements in the deep learning domain and new architectures being developed using neural networks, the problem of semantic understanding has been taken care of to a certain extent. All the methods and classifiers discussed here have their own advantages and disadvantages. The correct choice of the classifier with regard to the data that needs to be handled along with the right preprocessing technique will decide the effectiveness of the whole process. The idea in this paper can also be extended to the development of a good disease tagger.

References

1. Klosowski, P. (2018). Deep Learning for Natural Language Processing and Language Modelling. 2018 Signal Processing: Algorithms, Archi- tectures, Arrangements, and Applications (SPA).
2. Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. IEEE Computational Intelligence Magazine, 13(3), 55–75.
3. L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, “Deep learning for answer sentence selection,” Computing Research Repository (CoRR), vol. abs/1412.1632, 2014.
4. Y. Bengio, and S. Bengio, ”Modeling High-Dimensional Discrete Data with Multi-Layer Neural N etworks.” pp. 400-406, 1999.
5. Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, ”A neural prob- abilistic language model,” The Journal ofMachine Learning Research, vol. 3, pp. 1137-1155, 2003.
6. T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, ”Recurrent neural network based language model.” 11 th Annual Confer- ence of the International Speech Communication Association, pp. 1045-1048,2010.
7. T. Mikolov, S. Kombrink, L. Burget, J. H. Cernocky, and S. Khudanpur, ”Extensions of recurrent neural network language model.” Acoustics, Speech and Signal Processing, 2011 IEEE International Conference on. IEEE, pp. 5528-5531,2011.
8. H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, ”Structured output layer neural network language model.” Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on. IEEE, 2011.
9. H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, ”Structured output layer neural network language models for speech recognition,” Audio, Speech, and Language Processing, IEEE Transactions on, vol.21, no. 1, pp. 197-206,2013.
10. Fu-Lian Yin, Xing-Yi Pan, Xiao-Wei Liu, and Hui-Xin Liu. (2015). Deep neural network language model research and application overview. 2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP).
11. Hochreiter, Sepp, and Jrgen Schmidhuber. ”Long short-term memory.” Neural computation 9.8 (1997): 1735-1780
12. Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena, 132306.
13. Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In International Conference on Machine Learning, pages 1310–1318, 2013.
14. Oana Frunza, Diana Inkpen, and Thomas Tran, Member, IEEE, “A Ma- chine Learning Approach for Identifying Disease-Treatment Relations in Short Texts”, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 6, JUNE 2011.
15. Potdar, K., S., T., and D., C. (2017). A Comparative Study of Cat- egorical Variable Encoding Techniques for Neural Network Classi- fiers. International Journal of Computer Applications, 175(4), 7–9. doi:10.5120/ijca2017915495.
16. Alshari, E. M., Azman, A., Doraisamy, S., Mustapha, N., and Alkeshr, M. (2017). Improvement of Sentiment Analysis Based on Clustering of Word2Vec Features. 2017 28th International Workshop on Database and Expert Systems Applications (DEXA).

17. Suleiman, D., and Awajan, A. (2018). Comparative Study of Word Embeddings Models and Their Usage in Arabic Language Applications. 2018 International Arab Conference on Information Technology (ACIT).

18. I. Goodfellow, Y. Bengio, A. Courville. Deep Learning. MIT Press, 2016.

19. H. Bourlard, Y. Kamp. Auto-association by multiplayer perceptrons and singular value decomposition. Biological Cybernetics, 1988, 59:291-294.

20. G. E. Hinton, R. S. Zemel. Autoencoders, minimum description length, and Helmholtz free energy. Advances in Neural Information Processing Systems 6. J. D. Cowan, G. Tesauro and J. Alspector (Eds.), Morgan Kaufmann: San Mateo, CA.

21. Prechelt, L. (2012). Early Stopping — But When? Neural Networks: Tricks of the Trade, 53–67.