ISSN 2278-3091
Volume 8, No.4, July – August 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse106842019.pdf
https://doi.org/10.30534/ijatcse/2019/106842019

Development of Topic Modeling Framework Using Probabilistic Recurrent Neural Network

P.Lakshmi Prasanna[1], Dr.D.Rajeswara Rao[2]
[12]Koneru Lakshmaiah Education Foundation, India, lakshmiprasannap87@gmail.com

ABSTRACT

A topic model is a probability based model that finds the collection of documents. The basic concept is to treat the documents as combinations of topics in the topic model, and each topic is viewed as a probability distribution of the. In this paper we proposed LDA Algorithm using Probabilistic recurrent neural network (PRORNN) to classify the text documents. Topic modeling refers to the task of Discovering Latent Topics in the text corpus set, where the output is commonly represented as top terms appearing in each topic. This algorithm , probabilistic recurrent neural network (PRORNN) is implemented first with 2 News groups data set and later with 20 News groups dataset and all the results are tabulated. The performance of PRORNN algorithm was compared with the state of art of algorithms for topic classification.

Key words: Topics, Tokens, corpus, Terms, unstructured Data, frequency, stemming, Text

1. INTRODUCTION

Text Mining refers to the process of extracting high-quality information from a large amount of unstructured text using computational methods and techniques[1,2,3]. Unstructured data are ubiquitous and can be in forms such as new articles, books, and social media. The amount of unstructured text data is growing rapidly, and Computer World magazine declares that unstructured information might be more than 70%-80% of all data in organizations. text mining is relevant to enable the effective and efficient use of huge quantities of text[5,6,7]. **T**able 1 and figure 1 shows sources of unstructured data AND figure 2 shows the architecture of topic modeling using probabilistic recurrent neural network.

1.1 Sources of Unstructured Data

Table 1: Sources of Unstructured Data

Source	Example
Social Media	Facebook, LinkedIn, Google+, Instagram, YouTube
Location/Geo Data	GPS, Weather, traffic
Machine-generated/Sensor	Call Detail Records, weblogs, smart meters, manufacturing sensors, equipment logs or digital exhaust, trading systems, data records
Digital Streams	Video, audio, and images
Text Documents	Email, PowerPoint, Spreadsheets, Word-processing
Logs	File Log, Clickstream
Transactions	customer information from CRM systems, web store, general ledger, transactional ERP
Micro-Blogging	Twitter, Customer feedback streams

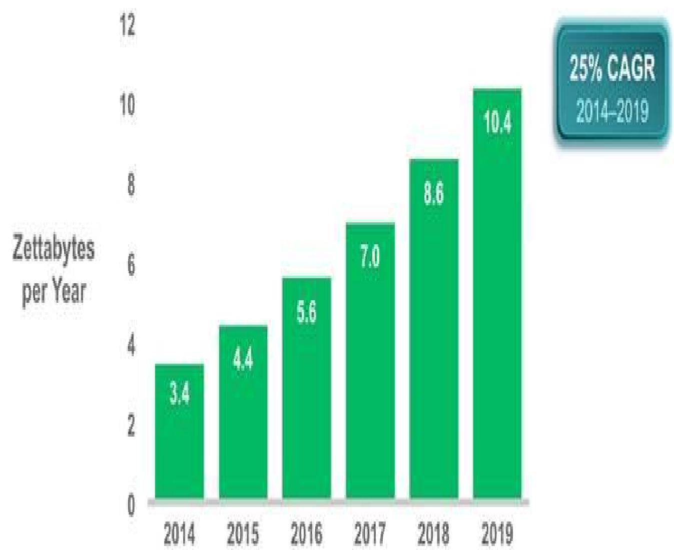


Figure 1: Annual data growth of Unstructured Data from 2014–2019[12].

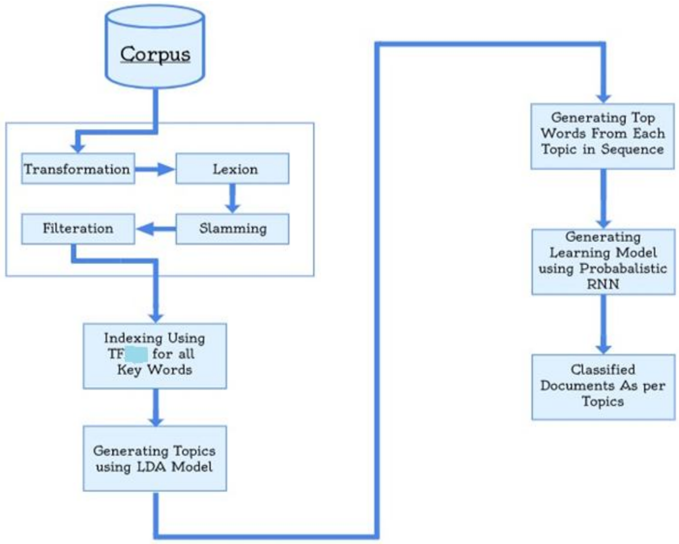


Figure 2: Architecture of Topic Modeling Using Probabilistic Recurrent Neural Network.

1.2 Data pre-processing

The proposed Topic modeling analysis starts with pre-processing the data obtained from 20 news group's data set[29,30]. A document which is defined as a sequence of words and punctuations refers to the data set. I experimented on 20 news group data set and 2 news group data set and two classes consists of 2000 documents & twenty classes consists of 18,846 documents are identified in pre processing. in a pre-processing we applied NLP Techniques that are 1.convert uppercase to lowercase ,2.removing special characters and dividing tokens 3.remove stop words 4.stemming 5.Apply the Term Frequency 6.Finally to create Document Term Matrix[25,27,4].In the Results it shows in the figures 3,4.

The data Pre processing the following steps:

- All White Spaces are removed in the Documents
- All Special Characters are to be removed
- All words are divided into tokens by using tokenization
- All these words to apply stemming process
- All the words to apply lemmatization concept
- Apply the Term Frequency (TF) for all words.
- By apply all these things we can get words and these words will store it document term matrix.

Algorithm for to Preprocess the data and generate document vector Model.

1. To create Directory for to Store Corpus=~ fdata
2. Read Whitespaces and to Remove whitespaces from fdata
3. To Remove stopwords= "English"
4.def stopwords (fdata):
5. fdata2=[]
6. fdata=nlp(fdata)
7. for token in fdata:
8. If (token.is_stop == False) & (token.pos_ !="PUNCT"):
9. new.append (token.string.strip ())
10. fdata2=" ".join (str(token) for token in fdata2)
11. Return fdata2
function to lemmatize the tokens
12. def lemmatize (fdata):
13. fdata2=nlp (fdata)
14. fdata2=""
15.for tokens in fdata:
16. fdata2+=" "+token.lemma_
17. Return nlp (fdta2)
#vectorising the sentences
18.def dtm (sent, fdata2):
19. dtm = np.zeros(200)
20. Numtokens = 0
21. for tokens in sent.split:
22. dtm = np.add(dtm, model [str(tokens)])
23. Numtokens+=1
24 return dtm

Document Term Matrix for 2 groups Data

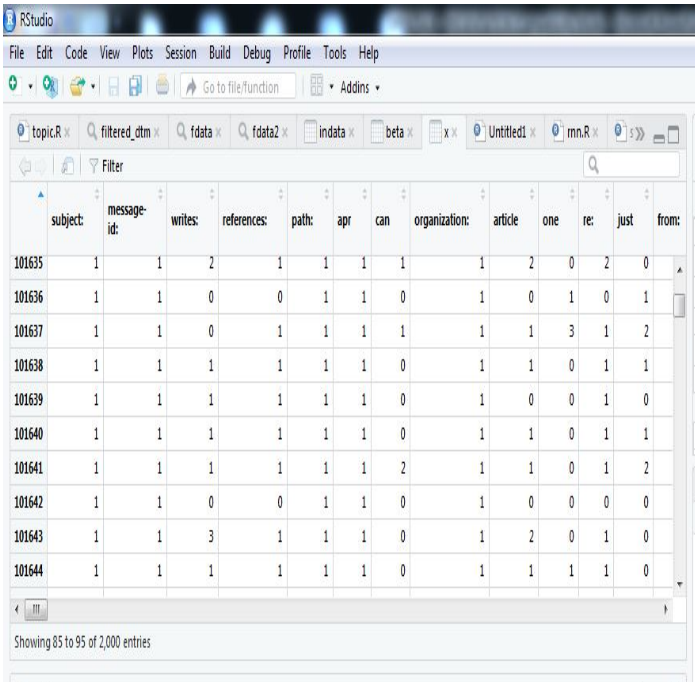
```
<<DocumentTermMatrix (documents: 2000, terms: 52948)>>
Non-/sparse entries: 276147/105619853
Sparsity           : 100%
Maximal term length: 311
Weighting          : term frequency (tf)
```

Figure 3: Document Term Matrix for 2 groups Data

After Applying TF

```
<<DocumentTermMatrix (documents: 2000, terms: 4140)>>
Non-/sparse entries: 180532/8099468
Sparsity           : 98%
Maximal term length: 193
Weighting          : term frequency (tf)
```

Figure 4: Bag of Words of Data Document Term Matrix (dtm)



	subject:	message-id:	writes:	references:	path:	apr	can	organization:	article	one	re:	just	from:
101635	1	1	2	1	1	1	1	1	2	0	2	0	
101636	1	1	0	0	1	1	0	1	0	1	0	1	
101637	1	1	0	1	1	1	1	1	1	3	1	2	
101638	1	1	1	1	1	1	0	1	1	0	1	1	
101639	1	1	1	1	1	1	0	1	0	0	1	0	
101640	1	1	1	1	1	1	0	1	1	0	1	1	
101641	1	1	1	1	1	1	2	1	1	0	1	2	
101642	1	1	0	0	1	1	0	1	0	0	0	0	
101643	1	1	3	1	1	1	0	1	2	0	1	0	
101644	1	1	1	1	1	1	0	1	1	1	1	0	

Figure 5: Vector Representation of Data

Generate Semantic Concepts of Documents:

For Finding Topics we used LDA Algorithm by Applying Bayes theorm to calculate Probability and topic identification of Each term &Each Document. For Feature Reductions to use LDA Algorithm to reduce features and filtered top terms of each topic and store it in the Filtered Document Term Matrix[28,15].

Latent drichlent allocation (LDA) is mainly used in analyzing text documents .it assumes that there N topics according to documents are generated and each topic is represented by multinomial distribution over y words in the vocabulary [8,9,10]. A document wd={wdt}dtt=1 is generated by sampling a mixture and these topics and sampling of words from the mixture.

A process of LDA is as follows

1 .the each topic n=1,2,3....N Draw a word proportion Φn~drichlet (β)

2. For each document d=1,2...D Draw topic proportions θd ~drichlet (α)

3. For each word t=1,2.....dt

Draw a topic assignment pdt~catogorical (θd) Draw a word wdt~catogorical (ΦPdn)[19,20,21]

In this algorithm the first step it shows that number of topics and the second step represents every word temporary allocates to topics and this process done randomly and sometimes same words may be applied to different topics [11, 13,14]. The third step shows that update the topic assignment based on their probability based on the two criteria's:

1. The first criteria is how prevalent is that word across the topics it can be termed as P(w/t)[17].

2. The second criteria is how prevalent are topics in the document P (t).in figure 5,6,7,8 shows that top terms, and probabilities of each term.

```
top5termsperTopic
      Topic 1               Topic 2
[1,]  "subject:"            "the"
[2,]  "message-id:"         "newsgroups:"
[3,]  "writes:"             "lines:"
[4,]  "references:"         "gmt"
[5,]  "path:"               "date:"
[6,]  "apr"                 "from:"
[7,]  "can"                 "1993"
[8,]  "organization:"       "re:"
[9,]  "article"             "organization:"
[10,] "one"                 "people"
[11,] "re:"                 "apr"
[12,] "just"                "article"
```

Figure 6: Top terms per topics for 2Groups Data

Top Terms with Probabilities for 2 Groups Data

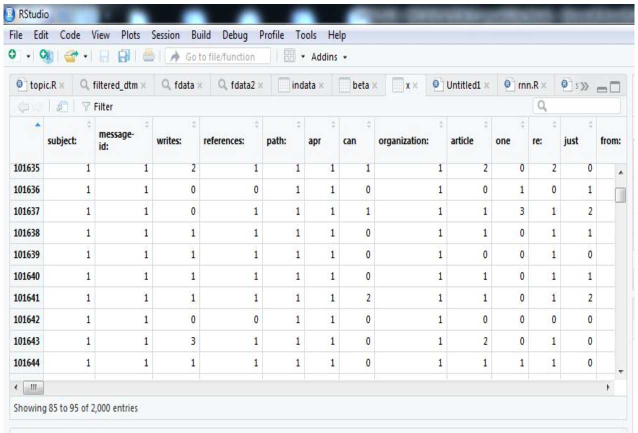
```
probabilities
      subject:          message-id:          writes:
  0.0172949546         0.0170884915     0.0135671864
   references:                path:              apr
  0.0122303777         0.0118613931     0.0114458514
           can        organization:          article
  0.0112066146         0.0088592080     0.0075160422
           one                  re:             just
  0.0070988660         0.0067745911     0.0064550337
         from:                date:              car
  0.0063414342         0.0062106215     0.0061137902
   alt.atheism                 like             know
  0.0060162284         0.0049348162     0.0048697613
          will     nntp-posting-host:            see
  0.0047899506         0.0046996792     0.0045395942
```

Figure 7: Top Terms with Probabilities for 2 Groups Data

Top Terms with Probabilities for 20 news Groups Data

```
Topic: 0
Words: 0.022*"encrypt" + 0.020*"chip" + 0.015*"secur" + 0.015*"clipper" + 0.010*"key" + 0.009*"public

Topic: 1
Words: 0.020*"caltech" + 0.020*"keith" + 0.014*"berkeley" + 0.012*"moral" + 0.009*"iastat" + 0.008*"a

Topic: 2
Words: 0.014*"islam" + 0.013*"moral" + 0.010*"object" + 0.007*"valu" + 0.007*"human" + 0.006*"attack"

Topic: 3
Words: 0.006*"stratus" + 0.005*"udel" + 0.005*"happen" + 0.004*"auto" + 0.004*"hand" + 0.004*"batf" +

Topic: 4
Words: 0.029*"space" + 0.025*"nasa" + 0.011*"orbit" + 0.009*"launch" + 0.007*"moon" + 0.007*"earth" +
```

Figure 8: Top Terms with Probabilities for 20 news Groups Data



	subject:	message-id:	writes:	references:	path:	apr	can	organization:	article	one	re:	just	from:
101635	1	1	2	1	1	1	1	1	2	0	2	0	
101636	1	1	0	0	1	1	0	1	0	1	0	1	
101637	1	1	0	1	1	1	1	1	1	3	1	2	
101638	1	1	1	1	1	1	0	1	1	0	1	1	
101639	1	1	1	1	1	1	0	1	0	0	1	0	
101640	1	1	1	1	1	1	0	1	1	0	1	1	
101641	1	1	1	1	1	1	2	1	1	0	1	2	
101642	1	1	0	0	1	1	0	1	0	0	0	0	
101643	1	1	3	1	1	1	0	1	2	0	1	0	
101644	1	1	1	1	1	1	0	1	1	1	1	0	

Showing 85 to 95 of 2,000 entries

Figure 9: Filtered Document Term Matrix (fdtm)

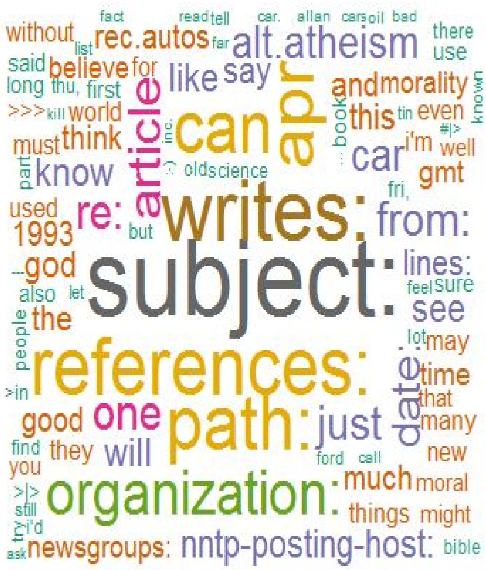


Figure 10: Word Cloud of the Data

Probabilistic Recurrent Neural network Topic Model

For Feature Reductions to used Lda Algorithm, to reduce features and filtered top terms of each topic and store it in the Filtered Document Term Matrix [18,28,16]. These reduced Terms to take it as input to the probabilistic Recurrent Neural Network and the top terms probability values as weights of network and used hidden layers as LSTM Layers and to calculate binary cross entropy for finding error loss and Adam optimizer is used to update the networks weights iterative based on training data and finally to classify documents based on topic and to calculate the Accuracy of the Network[22,23,24]. In figure 9,10 shows the top words of the documents .

Algorithm for Probabilistic Recurrent Neural Network (PRORNN)

PRORNN is a probability based Model and for a document containing the words in fdtm.

1. Represent fdtm as vector form

   fdtm~w,Tt

2. Represent output variable as Y and to specify the range 0 to n documents.
3. Represent the input fdtm where dimensions as n to p.
4. Set Labelled as document 1 to q as t0

   1,q=~ t0

5 set labeled as document q+1 to n as t1

q+1, n=~ t1

For Training the network

1. To set as ή as j and hidden states (lstm layers) are ht,ht-1
2. Set the no of epochs are E.
3. Generate the Error Rate

For Testing

1. for testing to specifying Range of documents s to r.
2. Plot Documents in the Topic Wise[25]. The notations of PRORNN. Shows in table 2.

Notations of PRORNN

Table 2: Notations of PRORNN.

Symbol	Description
dtm	Document Term Matrix
f_{dtm}	Filtered Words in the Document Term Matrix
T_t	Top terms of the Documents
X,Y	Input and Output variable
n	No of Documents
p	Range Specified as Input
$t_0\ t_1$	Topic Assignments
s,r	Set of Range of Documents
ή	Learning Rate
E	No of Epochs

Activation functions: The sigmoid activation function It can add non-linearity to the output and returns a binary value of 0 or 1. This binary relationship of data can be computed by the sigmoid activation function:

$$Y = \frac{1}{1+e^{-w}}$$

Adam Optimization

The Adam optimization algorithm is an extension to stochastic gradient descent that is used in vision and natural language processing. Adam is an optimization algorithm that can used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

Binary Cross entropy: In Binary Cross Entropy is used to calculate the probability of words. where w is number of Words and Y is the Number of Classes and log is natural Logarithm, t ,i,j is 1 if word i is in class j and 0 otherwise, and P,i,j is the Predicted word i is in class j.

$$\text{Binary Cross Entropy} = -\frac{1}{w}\sum_{i=1}^{w}\sum_{J=1}^{Y} t_{i,j}\log(P_{i,j})$$

Accuracy:

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has defined that

$$\text{Accuracy} = \frac{\text{No of Correct Predictions}}{\text{Total Number of Predictions}}$$

Trained Epochs

```
Trained epoch: 1 - Learning rate: 0.6
Epoch error: 0.22413260830237
Trained epoch: 2 - Learning rate: 0.6
Epoch error: 0.0623705445027698
Trained epoch: 3 - Learning rate: 0.6
Epoch error: 0.0390665946306848
Trained epoch: 4 - Learning rate: 0.6
Epoch error: 0.0274639539940824
Trained epoch: 5 - Learning rate: 0.6
Epoch error: 0.0222470412120581
Trained epoch: 6 - Learning rate: 0.6
Epoch error: 0.0191627249097986
Trained epoch: 7 - Learning rate: 0.6
Epoch error: 0.0152465553434716
Trained epoch: 8 - Learning rate: 0.6
Epoch error: 0.0135154498748832
Trained epoch: 9 - Learning rate: 0.6
Epoch error: 0.0116343564033731
Trained epoch: 10 - Learning rate: 0.6
Epoch error: 0.0104846037631226
```

Figure 11: Trained Epochs on PRORNN

Error Rate Based on Epochs

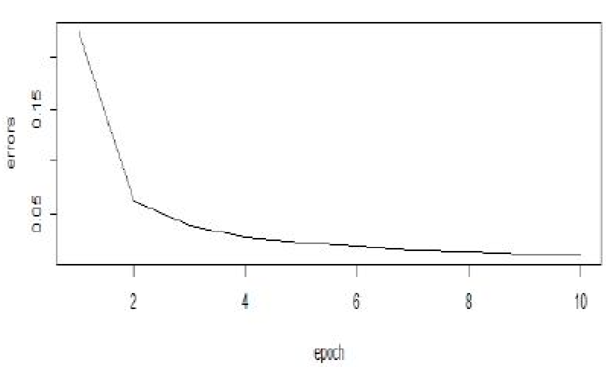


Figure 12: Error Rates Based on Epochs.

Classified Documents as Per Topics (2 groups Data)

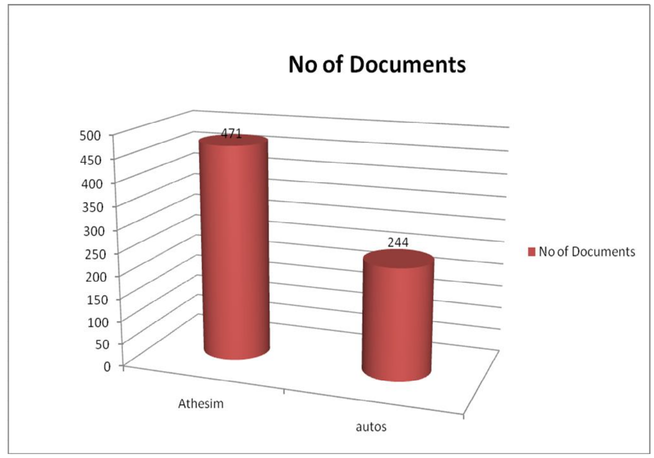


Figure 13: Classified Documents as Per Topics (2 groups Data)

Classified Documents as Per Topics (20 groups Data)

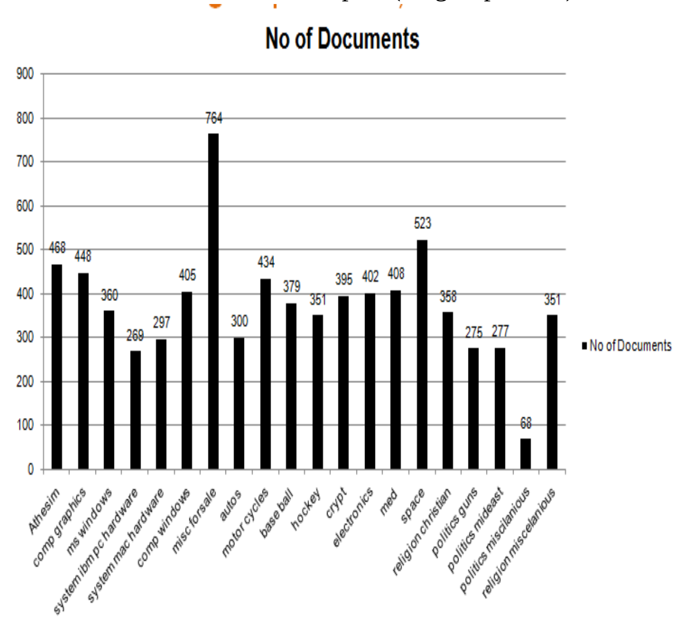


Figure 14: Classified Documents as Per Topic (20 groups Data)

In figure 11,12 shows the Error rate and figure 13,14 shows the classified documents as per topics.

Compare the results with Different Data Mining Algorithms and probabilistic recurrent neural network model[26,25].

Table 3: Comparison of results with Different Data Mining Algorithms and probabilistic recurrent neural network model.

Data Mining Algorithms	Accuracy
Naïve Bayes	71.4
Support Vector Machines	72.6
Probabilistic Recurrent Network	92.3

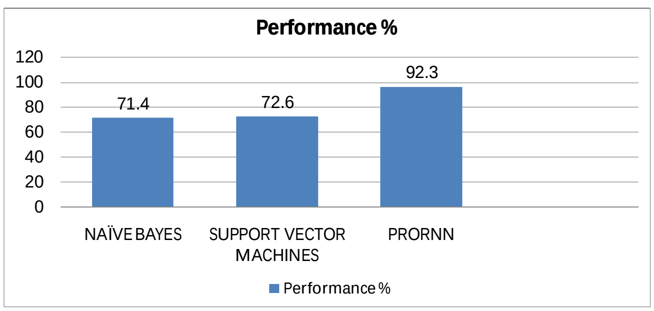


Figure 15: Accuracy of Different Data Mining Algorithms and probabilistic recurrent neural network model.

In table 3 shows the comparison of different data mining algorithms with PRORNN and figure 15 shows the accuracy of all the algorithms.

5. CONCLUSION AND FUTURE WORK

We proposed a Method for topic modeling using Probabilistic recurrent neural network .This Model Predict the topics for Documents based on probability values. This algorithm is useful for classification of documents where semantic meaning of terms is to be considered. As we applied LDA for reducing the terms, the complexity of learning model decreased and the accuracy of Probabilistic recurrent neural network is increased .The documents can be classified topic wise easily and there is no loss of information due to memory inconsistency as it can remember large words also. This work can be extended to predict authors & Topics from Documents.

REFERENCES

1. Sh sajid, "Automatic Topic Modeling for Single Document Short Texts", International Conference on Frontiers of Information Technology ,2017.
2. Jennifer Sleeman, Milton Halem, Tim Finin ,"Discovering Scientific Influence using Cross-Domain Dynamic Topic Modeling", IEEE International Conference on Big Data (BIGDATA) ,2017
3. Xiaoping Sun ,"Textual Document Clustering using Topic Models" Tenth International Conference on Semantics, Knowledge and Grids, 2014.
4. S. M. Weiss, N. Indurkhya, T. Zhang, and F. Damerau, "Text mining: predictive methods for analyzing unstructured information". Springer Science and Business Media, 2010.
5. W. Fan, L. Wallace, S. Rich, and Z. Zhang, "Tapping the power of textmining,", Communications of the ACM, 2006,vol. 49, no. 9, pp. 76–82. https://doi.org/10.1145/1151030.1151032
6. B.Meena Preethi,Dr.P.Radha,"A Survey Paper on Text Mining - Techniques, Applications And Issues", Next Generation Computing Technologies,2012
7. W.He,"Examining students online interaction in a live video streaming environment using data mining and text mining", Computers in Human Behavior, 2013,vol. 29, no. 1, pp. 90–102.
8. David Michael andrzejewski,"Incorporating domain knowledge in latent topic models ", 2010
9. Hanna M.Wallach ,"structured topic models for language" ,2008
10. Boyd-graber and Jordan ,"Linguistic Extensions of topic models" , 2010
11. Brandon Malone ,"Topic models ",2014
12. Adanma Cecilia Eberendu,,"Unstructured Data: an overview of the data of Big Data", International Journal of Computer Trends and Technology , 2016,Vol-38,no-1,
13. David M.Blei,Andrew Y,"Latent Drichlent Allocation "2003
14. . Hamed Jelodar · Yongli Wang · Chi Yuan · Xia Feng · Xiahui Jiang · Yanchao Li · Liang Zhao ,"Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey ".
15. Blei, D.M., A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation. Journal of machine Learning research", 2003. Vol-3,no –Jan, p. 993-1022.
16. David M. Blei ,"Surveying a suite of algorithms that offer a solution to managing large document archives",2012

17. Amir Karami Aryya Gangopadhyay Bin Zhou Hadi Kharrazi ,"Fuzzy Approach Topic Discovery in Health and Medical Corpora", 2013
18. A Zhou Tong,Haiyi Zhang, "Text Mining Research Based On Lda Topic Modelling" ,2016
19. Kaveh Bastani1 , Hamed Namavari, "Jeffry Shaffer Latent Dirichlet Allocation (LDA) for Topic Modeling of the CFPB", Consumer Complaints,2016
20. Rubayyi Alghamdi, Khalid Alfalqi,"Comparative Text Analytics via Topic Modeling in Banking".
21. Yu chen, Rhaad M.Rabbani,aparna gupta,mohammad j zaki ,"comparative text analytics via topic modeling in banking".
22. PreetChandan Kaur, TusharGhorpade, Vanita Mane, "Extraction of Unigram and Bigram Topic List by using Latent Dirichlet Markov Allocation and sentiment Classification ".
23. Rizwana .S,,Challa. K.S.Rafi ,S.S.Imambi ,
24. "Enhanced biomedical data modeling using unsupervised probabilistic machine learning technique", International Journal of Recent Technology and Engineering, Vol- 7, no- 6, March 2019, Pages 579-582
25. 24. A Neustein, SS Imambi, M Rodrigues ,"Application of text mining to biomedical knowledge extraction: analyzing clinical narratives and medical literature A Teixeira", L Ferreira, DeGyter publication, 2014
26. 25. P. Lakshmi Prasanna, D. Rajeswara Rao,"Design and Development of Topic Modeling for Probabilistic Recurrent Neural Network", International Journal of Engineering and Advanced Technology, Vol-8,no-5, June 2019.
27. .P. Lakshmi Prasanna, D. Rajeswara Rao,"Probabilistic Recurrent Neural Network for Topic Modeling", International Journal of Innovative Technology and Exploring Engineering, Vol-8 ,no-4, February 2019.
28. 27 P. Lakshmi Prasanna, D. Rajeswara Rao ,"A Text Mining Research Based On Topic Modeling using Latent Dritchlent Allocation", International Journal of Recent Technology and Engineering, Vol-7,no-5, January 2019.
29. Ibtihal S. Makki, Fahad Alqurashi ,"An Adaptive Model for Knowledge Mining in Databases "EMO_MINE for Tweets Emotions Classification", International Journal of Advanced Trends in Computer Science and Engineering, Vol-7, No.3, May- June 2018. https://doi.org/10.30534/ijatcse/2018/04732018
30. Priyanka Thakur ,Dr. Rajiv Shrivastava,"A Review on Text Based Emotion Recognition System", Vol-7, No.5, September - October 2018. https://doi.org/10.30534/ijatcse/2018/01752018