Tribhuvan University Journal
Vol. 33, No. 1: 11-22, June, 2019
Centre for Research, Tribhuvan University,
Kathmandu, Nepal
DOI: https://doi.org/10.3126/tuj.v33i1.28677

NEPALI TEXT DOCUMENT CLASSIFICATION USING DEEP NEURAL NETWORK

Sanjeev Subba[1], Nawaraj Paudel[2], Tej Bahadur Shahi[2]

ABSTRACT

An automated text classification is a well-studied problem in text mining which generally demands the automatic assignment of a label or class to a particular text documents on the basis of its content. To design a computer program that learns the model form training data to assign the specific label to unseen text document, many researchers has applied deep learning technologies. For Nepali language, this is first attempt to use deep learning especially Recurrent Neural Network (RNN) and compare its performance to traditional Multilayer Neural Network (MNN). In this study, the Nepali texts were collected from online News portals and their pre-processing and vectorization was done. Finally deep learning classification framework was designed and experimented for ten experiments: five for Recurrent Neural Network and five for Multilayer Neural Network. On comparing the result of the MNN and RNN, it can be concluded that RNN outperformed the MNN as the highest accuracy achieved by MNN is 48 % and highest accuracy achieved by RNN is 63%.

Keywords: Neural Network, Text Classification, Bag of words, Deep Learning, Machine Learning

INTRODUCTION

Modern information age produces vast amount of textual data, which can be termed in other words as unstructured data. Internet and corporate spread across the globe produces textual data in exponential growth, which needs to be shared, on need basis by individuals. If data

1 Mr. Subba is a Master's student, Central Department of Computer Science and IT, Kathmandu, TU.

2 Mr. Paudel and Mr. Shahi are Lecturers, Central Department of Computer Science and IT, Kathmandu, TU.

generated is properly organized, classified then retrieving the needed data can be made easy with least efforts. Hence the need of automatic methods to organize, classify the text becomes essential. Automatic classification of text refers to assigning the documents to a set of pre-defined classes based on the textual content of the document (Song et al. 2005).

Ideally, in the text domain, the classifier should classify incoming documents to the right existing classes used in training and also detect those documents that don't belong to any of the existing classes. This problem is called open world classification or open classification (Geli & Bing 2015)

Automatic classification of text documents has a wide range of scope. It is helpful for news editor for classifying the incoming news into predefined classes. It is also useful for the classification of scientific text into predefined class such as biology, chemistry and zoology etc.

LITERATURE REVIEW

In 1943 Warren McCulloch and Walter Pitts created a computational model for neural networks based on mathematics and algorithms called threshold logic. This model paved the way for neural network research. They highlighted how neurons might work. In order to describe how neurons in the brain work, they modeled a simple neural network using electrical circuits (Pitts & McCulloch 1943).

In 1949, Donald Hebb wrote "the Organization of Behavior", a work which pointed out the fact that neural pathways are strengthened each time they are used, a concept fundamentally essential to the ways in which humans learn. If two nerves fire at the same time, he argued that the connection between them is enhanced (Hebb & DO 1949).

Rosenblatt created the perceptron, an algorithm for pattern recognition. With mathematical notation, Rosenblatt described circuitry not in the basic perceptron, such as the exclusive-OR circuit that could not be processed by neural networks at the time (Kussul, et al. 2001).

In 1962, Author developed a learning procedure that examines the value before the weight adjusts it (i.e. 0 or 1) according to the rule:

Weight Change = (Pre-Weight line value) * (Error / (Number of Inputs)).

It was based on the idea that while one active perceptron may have a big error, one can adjust the weight values to distribute it across the

network, or at least to adjacent perceptron. Applying this rule still results in an error if the line before the weight is 0, although this will eventually correct itself. If the error is conserved so that all of it is distributed to all of the weights than the error is eliminated. They developed learning procedure that examines the value before the weight adjusts (Werbos 1974).

Neural network research stagnated after machine learning research by Minsky & Papert in 1969. They discovered two key issues with the computational machines that processed neural networks. The first was that basic perceptron were incapable of processing the exclusive-or circuit. The second was that computers didn't have enough processing power to effectively handle the work required by large neural networks. Neural network research slowed until computers achieved far greater processing power (Minsky & Papert 2017). The first multilayered network was developed in 1975, an unsupervised network. The term Deep Learning was introduced to the Machine Learning community in 1986 (Dechter 1986). Support Vector Machines were applied to text classification (Joachims 1998). It was an binary classifier which separate the data using a concept of hyper-plane which maximize the distance between supporting hyper-plane.

AdaBoost was enhanced to handle multi-labels in 2000 by Schapire and Singer. They showed an approach where, the task of assigning multi-topics to a text is regarded as a ranking of labels for the text. This ranking-based evaluation was inspired by Information Retrieval (Singer & Schapire 2000).

In 2018, Meng, et al. in the paper "Weakly-Supervised Neural Text Classification" compared the classification performance of different neural classifiers. In their experiment they evaluated the empirical performance of their method for weakly supervised text classification. They proposed a weakly-supervised text classification method built upon neural classifiers. This work was significant as result shown that method outperforms baseline methods significantly, and it is quite robust to different settings of hyper-parameters and different types of user-provided seed information. They suggested to study on how to effectively integrate different types of seed information to further boost the performance of method (Meng et al. 2018).

In 2018, Neel Kant, Raul Puri, Nikolai Yakovenko, Bryan Catanzaro in paper "Practical Text Classification with Large Pre-Trained Language Models" .They compare the deep learning architectures of the Transformer and mLSTM and found that the Transformer outperforms the mLSTM

across categories. They demonstrated that unsupervised pre-training and fine-tuning provides a flexible framework that is effective for difficult text classification tasks. They found that the fine-tuning was especially effective with the transformer network (Kant et al. 2018).

Has, ImSak, Andrew Senior, Franc¸Oise Beaufays evaluated and compared the performance of LSTM RNN architectures on a large vocabulary speech recognition task – the Google Voice Search task. They used a hybrid approach for acoustic modeling with Long Short Term Memory (LSTM) RNNs, wherein the neural networks estimate hidden Markov model (HMM) state posteriors. They showed that Deep LSTM RNN architectures achieve state-of-the-art performance for large scale acoustic modeling. The proposed Deep LSTMP RNN architecture outperforms standard LSTM networks (Sak, Senior & Beaufays 2014).

It is seen form the literature review that the various deep learning techniques have been successfully applied to English text. However nobody has mentioned about Nepali text classification using deep learning approaches. Hence, it is natural to design and experiment the deep learning architecture for Nepali text classification problem.

RESEARCH QUESTIONS

This research work aimed to use deep learning technique to address the problem of classifying text documents automatically. The research questions are formulated as:

Are deep leering algorithms: Recurrent Neural Network (RNN) and Multi Neural Network (MNN) able to address the text classification problem for Nepali text.

How much accurately these methods will classify the Nepali text? What will be the best hyper-parameters for these methods at their best accuracy?

DATASET PREPARATION

The data set used in this work was collected from various online Nepali News portals using web crawler (Shahi & Pant 2018). The news portal namely *ratopati.com*, *setopati.com*, *onlinekhabar.com*, and *ekantipur.com* were used to gather text related to different news types. There were around 20 different classes of news data but only Business and Interview

class data has been used because it has the largest size compare to other news data in the dataset.

PREPROCESSING

Unstructured textual data usually requires some preprocessing before it can be analyzed. This process includes a number of optional text preprocessing and text cleaning steps, such as replacing special characters and punctuation marks with spaces, removing duplicate characters, removing user-defined or built-in stop-words, and word stemming (Uysal & Gunal, 2014). This process cleans the data and then fed into another step of processing in pipeline of classification system.

DATA VECTORIZATION

Vectorization is the process to convert the raw data to the data which can be feed into the computer. There are different types of vectorization methods. For this research work, Bag of Words approach is used for vectorization. The bag-of-words model is commonly used methods of document classification where the occurrence of each word is used as a feature for training a classifier (Mahendran, Duraiswamy, Reddy, & Gonsalves, 2013). One approach which counts the occurrence of words is called bag of words. The basic idea is to take the word and count the frequency of the occurrence of those words from document. The word is considered as the feature and it is unique. For example:

Sentence 1: बैंकले फि, कमिसन, अन्य आम्दानी र विदेशी मुद्राको विनिमयबाट २३ करोड रुपैयाँ आर्जन गरेको छ.

Sentence 2: बैंकको निक्षेप ३१.३४ प्रतिशतले बढेर ६४ अर्ब ४८ करोड रुपैयाँ पुगेको छ.

The vector representation of sentences is shown in Table 1.

Table 1: Vector representation of sentences

Features	विदेशी	कमिसन	आम्दानी	आर्जन	बैंक
Sentence 1	1	1	1	1	1
Sentence 2	0	0	0	0	1

The vector representation of sentences is then feed into the neural network to train and test.

TEXT CLASSIFICATION SYSTEM

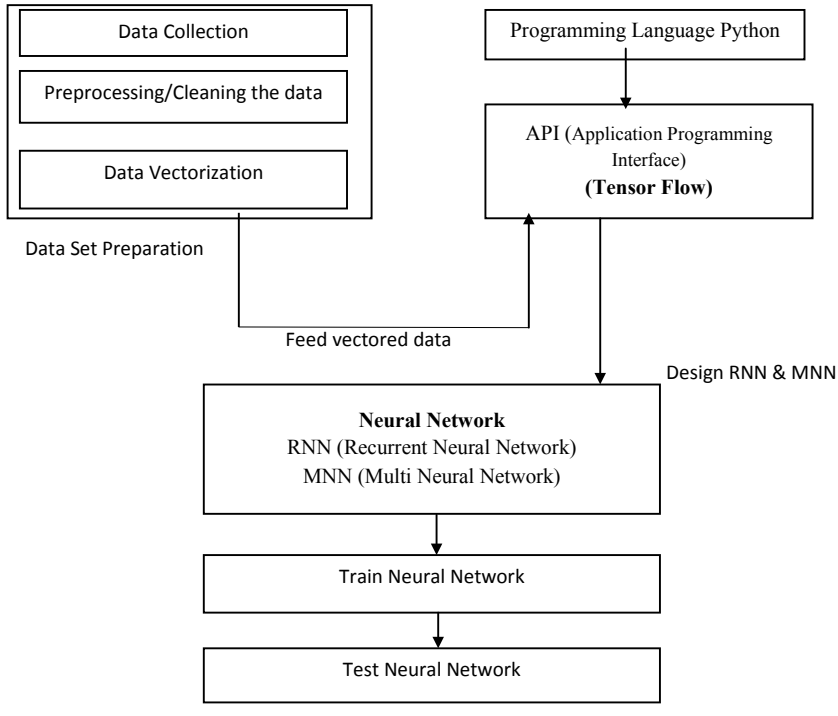

Figure 1: Discusses the steps taken to carry out this research.

As a first step in pipeline, the pre-processing which consists of removing noise such as HTML tags, special symbols etc. has been done. During these activities, tokenization of documents into words is also carried out. After preprocessing, the data encoding is done using vectorzation method. To create the vector for each text documents, we use a tf-idf calculation for each word and substitute this number in place of each word in the sentences. After vectorization, the main module of classification was designed using Multi-layer Perceptron Network and Recurrent Neural Network.

EXPERIMENTAL RUNS AND VALIDATIONS

Vectored Nepali text data are taken to train the Neural Network. Total sample data for the Nepali text is 33,880 documents. The optimal hyper-parameter like learning rate, epoch, batch size, hidden layer is chosen to conduct the different experiment (Leslie 2018).

Cross-fold validations

The total data sample of 33,880 documents in Business and Interview class were taken for the experiment purpose. This data set is further divided it to two set: training and testing in the ratio of 8:2. The resulting sample of data contains 27,104 text documents as training set and the number of text document in test set are 6,776. The parameter for evaluation used are accuracy, precision, recall and f-score (Powers 2011).

Observation on multi neural network (MNN)

The experiment runs from one to five in numbers and the hyper-parameter were adjusted to get the best result for the given architecture. The hyper-parameter as well as final loss found during the experiments were reported in the Table 4 for each experiment.

Table 4: Hyper-parameters and final loss for MNN

Experiment Number	Learning rate	Number of epoch	Batch size	Hidden layer	Hidden unit	Final loss
Experiment 1	0.01	10	100	3	100	410.62
Experiment 2	0.05	15	200	9	200	410.62
Experiment 3	0.09	20	300	12	300	410.62
Experiment 4	0.5	25	400	15	400	NAN
Experiment 5	0.9	30	500	18	500	NAN

NAN is shown whenever the loss is either too great or too small. Theoretically number is either close to – infinity or + infinity

The test experiment was carried out for Business and Interview class documents. 20% of total business class documents were feed to Multilayer Neural Network framework to evaluate the model's performance on four measures: Accuracy, Precision, Recall and F-score. The result of four experiment is shown in Table 5.

Table 5: The result of five experiments for MNN

Experiment Number	Accuracy	Precision	Recall	F-Score
Experiment 1	0.043	0.057	0.053	0.11
Experiment 2	0.421	0.384	0.415	0.391
Experiment 3	0.212	0.230	0.220	0.230
Experiment 4	0.483	0.590	0.480	0.53
Experiment 5	0.487	0.620	0.490	0.540

Validity and reliability testing for MNN experiments

The validity of experiments were measures in terms of number of documents accepted and rejected for a particular class by the classification system (Fawcett 2006). It is also represented in confusion matrix as shown in Table 6.

Table 6: The confusion matrix of five experiments for MNN

Experiment number	No of correctly classified item		No. of misclassified item	
	True positive	True negative	False positive	False negative
Experiment 1	194	102	3194	3286
Experiment 2	1301	1555	2087	1833
Experiment 3	807	633	2581	2755
Experiment 4	2032	1246	1356	2142
Experiment 5	2102	1204	1286	2184

Observation on recurrent neural network (RNN)

During the experimental run from one to five in numbers, the hyper-parameters were adjusted to get the best result for the given architecture. The hyper-parameter as well as final loss found during the experiments were reported in the Table 7 for each experiment.

The test experiment was carried out for Business class documents. 20% of total business class documents were feed to Recurrent Neural Network framework to evaluate the model performance on four measures: Accuracy, Precision, Recall and F-score. The result of five experiments is shown in Table 8.

Table 7: Hyper-parameters and final loss for RNN

Experiment number	Learning rate	Number of epoch	Batch size	Hidden layer	Hidden unit	Final loss
Experiment 1	0.01	10	100	3	100	262.26
Experiment 2	0.05	15	200	6	200	139.32
Experiment 3	0.09	20	300	9	300	92.4
Experiment 4	0.5	25	400	12	400	82.01
Experiment 5	0.9	30	500	18	500	102.01

Table 8: The result of five experiments for RNN

Experiment Number	Accuracy	Precision	Recall	F-Score
Experiment 1	0.59	0.51	0.61	0.56
Experiment 2	0.61	0.54	0.63	0.58
Experiment 3	0.63	0.55	0.65	0.60
Experiment 4	0.62	0.55	0.65	0.59
Experiment 5	0.58	0.52	0.59	0.55

Validity and reliability testing for RNN experiments

Table 9: The confusion matrix of five experiments for RNN

Experiment. number	No of correctly classified item		No. of misclassified item	
	True positive	True negative	False positive	False negative
Experiment 1	1752	2281	1636	1107
Experiment 2	1850	2321	1538	1067
Experiment 3	1880	2401	1508	987
Experiment 4	1878	2380	1510	1008
Experiment 5	1784	2203	1604	1185

The validity of experiments was measures in terms of number of documents accepted and rejected for a particular class by the classification system. It is also represented in confusion matrix as shown in Table 9.

DISCUSSION AND ANALYSIS OF RESULTS

The result of 5 experiments in test dataset with 20% of total data for MNN is shown in Figure 2. It shows the highest recall found in experiment 5 and lowest recall was on experiment 1. Lowest accuracy was in Experiment 1 and highest in Experiment 5. On average, performance evaluation parameter is good in experiment 5.

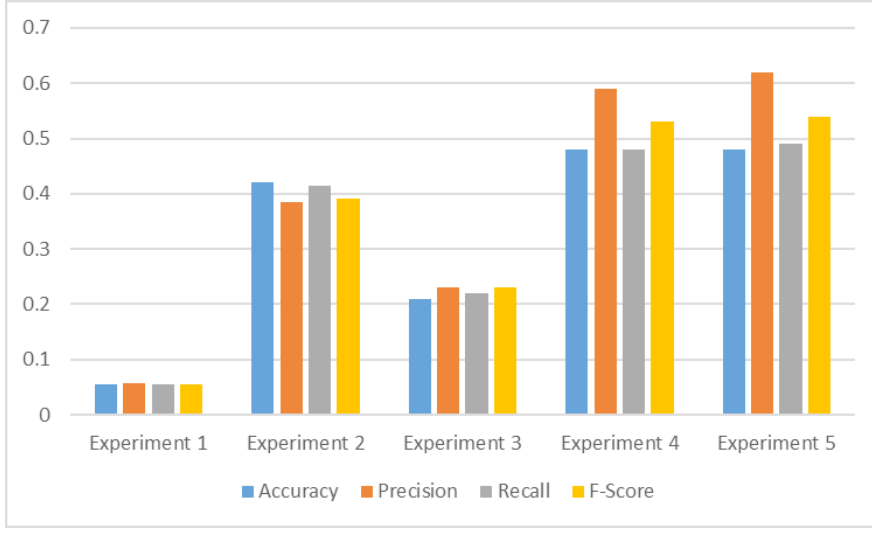

Figure 2: Bar chart for Nepali test data of MNN

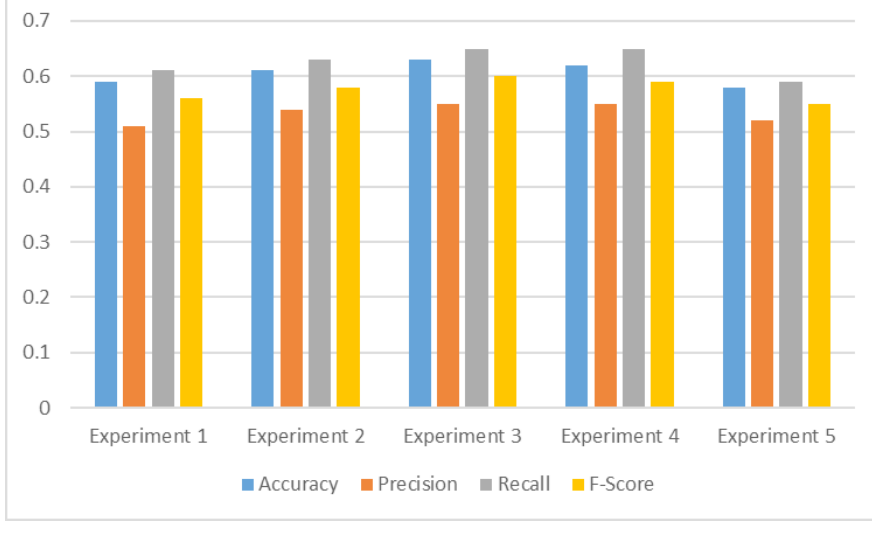

Figure 3: Bar chart for Nepali test data of RNN

Similarly, the experimental result for RNN is displayed in Figure 3. Testing with 20% sample data of Nepali data in RNN shows recall is lowest in experiment 1 and highest in experiment 3. Accuracy is lowest in experiment 1 and highest in experiment 3. On average, performance evaluation parameter is good in experiment 3.

On comparing the result of the MNN and RNN we can conclude that RNN outperform the MNN where the highest accuracy achieved by MNN is 48 % and highest accuracy achieved by RNN is 63%.

CONCLUSION AND RECOMMENDATIONS

Text Document classification was carried out with RNN and MNN neural network. The performance was examined on Nepali News Data. The experiment results showed the accuracy of RNN outperform the accuracy of MNN. The highest accuracy of MNN was 48% and the highest accuracy of RNN was 63%. The lowest accuracy of MNN was 4.3% and lowest accuracy of RNN was 59%.

Nepali data set test sample could not achieve the high accuracy in RNN and MNN because of the word embedding problem. The data collection should be large so the neural network model can train properly. Form the experiments, it can be recommended that lack of proper stemming and lemmatization technique in Nepali data set cause neural network model to fail to achieve high accuracy.

ACKNOWLEDGEMENT

Authors would like to acknowledge the Chinese academy of Science (CAS) for the providing grant to complete this work.

REFERENCES

Dechter, R. (1986). *Learning while searching in constraint-satisfaction problems.* University of California, Computer and Cognitive Systems Science Department, .

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters,* **27**(8): 861-874.

Geli, F. & Bing, L. (2015). Social media text classification under negative covariate shift. *Conference on Empirical Methods in Natural Language Processing.* Spinger, pp. 2347-2356.

Hebb, D. O. & DO, H. (1949). *The organization of behavior.* New York: Wiley.

Joachims, T. (1998). *Text categorization with support vector machines: Learning with many relevant features.* Springer, pp. 137-142.

Kant, N., Puri, R., Yakovenko, N. & Catanzaro, B. (2018). *Practical text classification with large pre-trained language models*. arXiv preprint arXiv:1812.0120.

Kussul, E., Kasatkina, T., Baidyk, T. & Lukovich, L. (2001). Rosenblatt perceptrons for handwritten digit recognition. *Proceedings of the International Joint Conference on Neural Networks.* IEEE, pp. 1516-1520).

Leslie, N. S. (2018). A disciplined approach to neural network hyper-parameters: Part 1. *CoRR*.

Mahendran, A., Duraiswamy, A., Reddy, A. & Gonsalves, C. (2013). Opinion mining for text classification. *International Journal of Scientific Engineering and Technology,* **2**(6): 589-594.

Meng, Y., Shen, J., Zhang, C. & Han, J. (2018). Weakly-supervised neural text classification. ACM, pp. 983-992.

Minsky, M. & Papert, S. A. (2017). *Perceptrons: An introduction to computational geometry.* MIT press.

Pitts, W. & McCulloch, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics,* **5**: 115-133.

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlatio. *Journal of Machine Learning Technologies, 2*(1), 37-63.

Sak, H., Senior, A. & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Fifteenth Annual Conference of the International Speech Communication Association.*

Shahi, T. B. & Pant, A. K. (2018). *Nepali news classification using naive bayes, support vector machines and neural networks.* International Conference on Communication Information and Computing Technology *(ICCICT)*. Mubai: IEEE, pp. 1-5.

Singer, Y. & schapire, R. E. (2000). Boos Texter: A boosting-based system for text categorization. *Machine Learning,* **39**: 135-168.

Song, M.-H., Lim, S.-Y., Kang, D.-J. & Lee, S.-J. (2005). Automatic classification of web pages based on the concept of domain ontology. *12th Asia-Pacific Software Engineering Conference (APSEC'05)*. IEEE, pp. 7-12.

Uysal, A. K. & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing and Management, 50*(1), 104-112.

Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD. Thesis, Harvard University.