Nepali Multi-Class Text Classification

Oyesh Mann Singh
Computer Science and Electrical Engineering
University of Maryland, Baltimore County
osingh1@umbc.edu
December 20, 2018

***Abstract*—Text classification is one of the most fundamental and researched task in the field of Natural Language Processing. Most of the paper discusses about the text classification in Nepali language using traditional machine learning algorithms. In contrast, this paper discusses about latest deep learning algorithm with latest word embedding, which produces better accuracy compared to those traditional algorithm.**

***Keywords*—NLP, text classification, Nepali news classification, deep learning, word2vec**

I. Introduction

Text classification is a broad field of study under text mining domain where researcher study about mapping a piece of text (news article, social media texts, documents) into predefined classes. Classification algorithms are Naïve Bayes, Nearest Neighbor, Support Vector Machine, Neural Network and their various improvised versions are generally used for the classification purpose.

Text classification is useful in various other applications like spam detection, topic labelling and sentiment analysis. Text data can be found anywhere like chats, email, review, social media, support tickets, movie reviews, survey response and more. The faster these texts can be understood, better the customer can be served resulting into higher customer satisfaction, eventually increased revenue generation for a company.

Hierarchical text classification is all about classifying the given documents or text into various labels and sub-labels which has parent-child relationship. Hence, the labels have hierarchical structure, however the given text can fall under multiple class. This is called Hierarchical Multi-Class Text Classification. With the trend of ever increasing data in large scale, there is also a greater need of automating data applications as quickly as possible so that people can make better decisions and analysis [1]. In this paper, we will be looking at comparison between various traditional machine learning algorithm to deep learning algorithms and its variations based on Nepali language.

In this paper, the documents are taken from different news sources using web crawler which are classified into 16 categories like Auto, Bank, Blog, Business Interview, Economy, Education, Employment, Entertainment, Interview, Literature, National News, Opinion, Sports, Technology, Tourism and World.

II. Literature Review

There has been a growing research interest in Nepali natural language processing domain specially in text classification lately. However, due to lack of standard corpus and open dataset in Nepal language, there has been a lag in research advancement in this particular language.

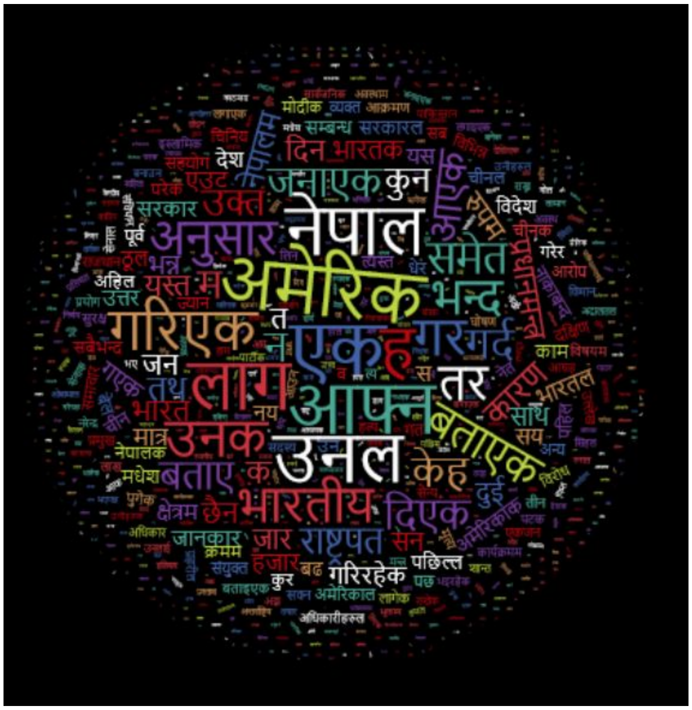

Figure 1. Word Cloud of train dataset

Lexicon pool augmented Naive Bayes classifier increases the accuracy in text classification[2], where they used self-created news documents to classify into 20 categories. Naive Bayes classifier was also used to detect spams in SMS in Nepali text[3]. [4] used neural networks and support vector machines to classify Nepali news documents in 20 categories where it argues Linear-SVM having highest accuracy compared to RBF-SVM, Naive-Bayes and Multi-Layered Perceptron. [5] performed Nepali document classification using SVM, however it summarizes using word2vec word embedding in Nepali language drastically increases F-score compared to when TF-IDF vectorizer is used.

[6] used convolutional neural network for sentence level classification with the word vector as input to the model of single convolution layer. [17] used character level convolution neural networks for text classification where the sequence of encoded characters were fed into model as input using one-hot encoding of fixed length. [11] used recursive neural

network for sentiment detection and sentence classification. [12] used hierarchical structure for document level sentiment classification where they used CNN or LSTM to get sentence representation from word representation and bi-directional gated recurrent neural network to get document representation.

In Neural Machine Translation domain, RNN based encoder-decoder model [7] suffers in accuracy as the length of the sentence increases. Since, [8] found out that due to encoder-decoder approach which encodes a whole input sentences into a fixed-length vector representation and decodes into the target sentence, became a bottleneck to translate the lengthy sentences. Attention model is all about giving more attention to the important part of text automatically compared to other parts, which helps in predicting the target sentence or labels with much better accuracy [8]. Attention model got more emphasis as it was able to show promising BLEU score even for lengthy sentences in machine translation.

Most of attention mechanism was still used in conjunction with recurrent network which limited computational efficiency because recurrent models typically factor computation along the symbol positions of input and output sequences [15]. New model architecture Transformer is developed in [15] which doesn't depend on recurrent or convolution network but dependent entirely on attention mechanism to draw global dependencies between input and output. Documents have hierarchical structure because words form sentences and sentences form documents. Therefore, if the neural network model, is created in hierarchical structure with attention mechanism along the word-level and sentence-level, it gives better accuracy [16].

III. Experiments

A. Dataset Preparation

Nepali corpus was provided by Dr. Bal[1] from Kathmandu University, Nepal. It is Nepali National Corpus which is also used in Sketch Engine[2]. The corpus had collection of documents from books, newspaper, journals and webtext. The corpus is Part-of-Speech tagged in XML format. The a new class[3] in python was written to read this XML format so that it is compatible with NLTK version of corpus reader, eventually we can directly load the corpus into word2vec embedding to create word representations.

The wordcloud in figure 1 represents the highest word frequencies visualization from the training dataset for the ”World” label.

The dataset was taken from web[4] which was prepared by scrapping Nepali news web documents and the files for each category were stored in their respective folders.

We can observe in figure 2 that total number of training records were 10000 and testing records were 5000. From figure 3, we can observe that the raw dataset is very much skewed towards the label National News compared to other label.

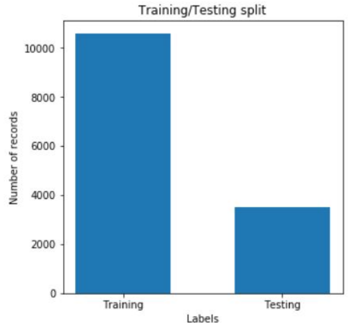

Figure 2. Separation of dataset into train and test

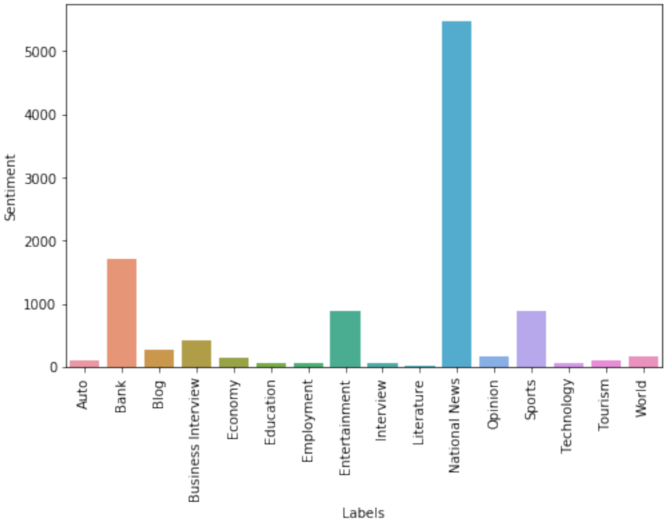

Figure 3. Distribution of dataset by labels

B. Text Preprocessing

The text preprocessing was done with the help of Nepali NLP Group[5], which has written nice blogs on Nepali language morphology, parsing, lemmatization, pos tagging, stemming and more. The text preprocessing code can be found here[6]

In the text preprocessing part, any kind of punctuation, symbols, numbers or any other invalid unicode characters were removed except hyphen which appears in between the words. After text cleaning, each documents were given a label and stored in csv format separated into training and testing dataset. This cleaned dataset and corpus was used to build vocabulary with different word representation packages like TF-IDF vectorizer and word2vec.

TF-IDF stands for Term Frequency Inverse Document Frequency, which also means the value of a word directly proportional to its frequency in a document however it is also inversely proportion to its frequency across the many documents. In other words, term frequency represents how often a particular word is repeating in a document and inverse document frequency reduces the value of word that appears a

[1] http://ku.edu.np/cse/faculty/bal/

[2] https://www.sketchengine.eu/

[3] https://github.com/oya163/oya-nepali-nlp/blob/master/NNCCorpus.py

[4] https://github.com/sndsabin/Nepali-News-Classifier

[5] http://nepalinlp.com/

[6] https://github.com/oya163/oya-nepali-nlp/blob/master/nepali-preprocessing.ipynb

lot across the document.

Word embedding is the vector representation of a specific words based syntactic and sematic word relationships. word2vec[18] is such one of the word embedding framework which are based on CBOW (Continuous Bag-of-Words) and SKIP-gram models. CBOW model inputs the context of each word and tries to predict the word corresponding to the context. SKIP-gram models takes in the target word to predict the context of the word.

C. *Machine Learning*

Various tradition machine learning algorithms like Logistic Regression, Support Vector Machine, Multinomial Naive Bayes, Bernoulli Naive Bayes, Nearest Neighbor, Perceptron, Multi-Layered Perceptron with (lbfgs/sgd/adam), Gradient Boosting Classifier, Bagging Classifier, SGD Classifier were used. In these algorithms, parameter tuning was performed using GridSearchCV and all of these algorithms used TF-IDF vectorizer.

Logistic regression was experimented with One Vs Rest Classifier schema along with L2 regularization, which took about 8 seconds achieving 68.64% accuracy on testing dataset. We can see in figure 4 the normalized confusion matrix based on the results produced by Logistic Regression. When GridSearchCV was used over Logistic Regression, it was found out that penatly term with 1000 was producing the best estimator having 77.145% accuracy. Similarly all other algorithms were ran using default parameters.

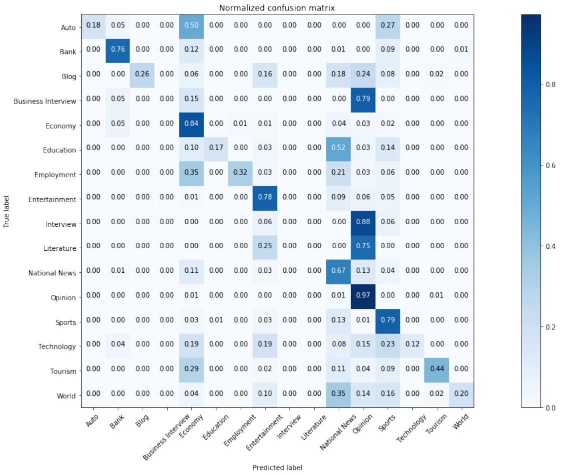

Figure 4. Confusion matrix for Logistic Regression

Deep learning algorithms like Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), Gated Recurrent Unit (GRU) and Adaptive GRUs were used. Adam optimizer was used for the optimization process with learning rate of 10^{-3}. They were trained for 40 epochs each. Figure 6 shows the accuracy of each deep learning algorithm where GRU has the highest accuracy. word2vec was used as word embedding. The loss plotting for other networks are in the github repository.

The code for these experiments can be found here[7]

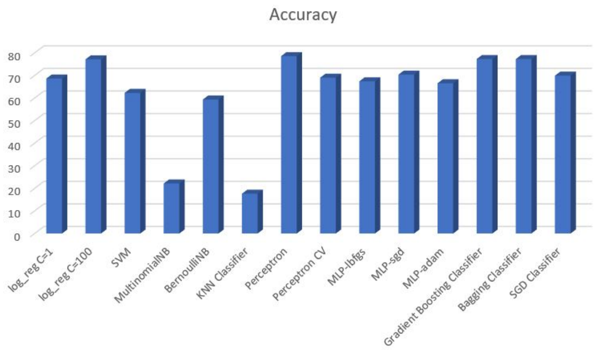

Figure 5. Comparison of various machine learning algorithms

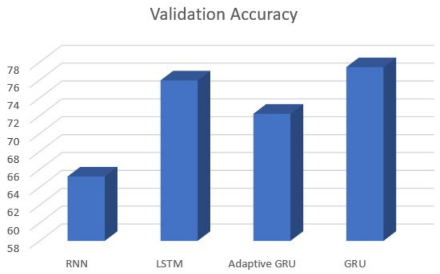

Figure 6. Comparison of various deep learning algorithms

IV. Future Work

- Collection of more data to increase the data size
- More rigorous text preprocessing (including lemmatization, stemming)
- Dataset balancing by merging similar labelled data like 'Bank', 'Business Interview', 'Economoy' into one category
- Usage of other word embeddings like Glove, fasttext
- Use of attention network

V. Conclusion

There should be more amount of data to train machine learning algorithms rigorously. The textual data in Nepali should be given more attention due to its complex morphology and language structure. Due to small size of dataset, deep learning algorithms was not able to perform better compared to traditional machine algorithms. For example, GRU network achieved 77.44% accuracy while simple perceptron was able to achieve 78.561% accuracy which was the highest among traditional machine learning algorithms.

Acknowledgment

I would like to thank Dr. Dutt for his guidance and sharing his knowledge on text classification using various machine learning algorithms and also for providing insightful comments on each research topic. I would also like to thank Dr. Bal for providing Nepali corpus.

[7] https://github.com/oya163/oya-nepali-nlp

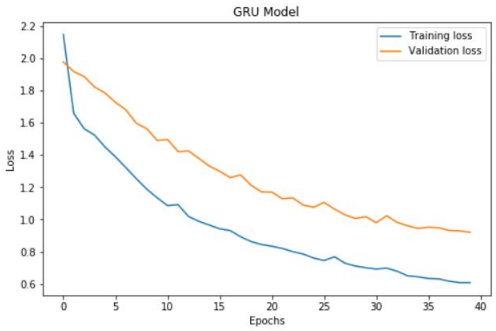

Figure 7. Loss of GRU network against the number of epochs

REFERENCES

[1] Thangaraj, M., and M. Sivakami. ”Text Classification Techniques: A Literature Review.” Interdisciplinary Journal of Information, Knowledge and Management 13 (2018): 117-136.

[2] Thakur, S. K., and Vivek Kumar Singh. ”A lexicon pool augmented naive bayes classifier for nepali text.” Contemporary Computing (IC3), 2014 Seventh International Conference on. IEEE, 2014.

[3] Shahi, Tej Bahadur, and Abhimanu Yadav. ”Mobile SMS spam filtering for Nepali text using naïve bayesian and support vector machine.” International Journal of Intelligence Science 4.01 (2013): 24.

[4] Shahi, Tej Bahadur, and Ashok Kumar Pant. ”Nepali news classification using Naïve Bayes, Support Vector Machines and Neural Networks.” 2018 International Conference on Communication information and Computing Technology (ICCICT). IEEE, 2018.

[5] Kafle, Kaushal, et al. ”Improving nepali document classification by neural network.” Proceedings of IOE Graduate Conference. 2016.

[6] Kim, Yoon. ”Convolutional neural networks for sentence classification.” arXiv preprint arXiv:1408.5882 (2014).

[7] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. ”Neural machine translation by jointly learning to align and translate.” arXiv preprint arXiv:1409.0473 (2014).

[8] Cho, Kyunghyun, et al. ”On the properties of neural machine translation: Encoder-decoder approaches.” arXiv preprint arXiv:1409.1259 (2014).

[9] Liu, Jingzhou, et al. ”Deep learning for extreme multi-label text classification.” Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2017.

[10] Pappas, Nikolaos, and Andrei Popescu-Belis. ”Multilingual hierarchical attention networks for document classification.” arXiv preprint arXiv:1707.00896 (2017).

[11] Socher, Richard, et al. ”Recursive deep models for semantic compositionality over a sentiment treebank.” Proceedings of the 2013 conference on empirical methods in natural language processing. 2013.

[12] Tang, Duyu, Bing Qin, and Ting Liu. ”Document modeling with gated recurrent neural network for sentiment classification.” Proceedings of the 2015 conference on empirical methods in natural language processing. 2015.

[13] Thangaraj, M., and M. Sivakami. ”Text Classification Techniques: A Literature Review.” Interdisciplinary Journal of Information, Knowledge and Management 13 (2018): 117-136.

[14] Tsaptsinos, Alexandros. ”Lyrics-based music genre classification using a hierarchical attention network.” arXiv preprint arXiv:1707.04678 (2017).

[15] Vaswani, Ashish, et al. ”Attention is all you need.” Advances in Neural Information Processing Systems. 2017.

[16] Yang, Zichao, et al. ”Hierarchical attention networks for document classification.” Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.

[17] Zhang, Xiang, Junbo Zhao, and Yann LeCun. ”Character-level convolutional networks for text classification.” Advances in neural information processing systems. 2015.

[18] Mikolov, Tomas, et al. ”Distributed representations of words and phrases and their compositionality.” Advances in neural information processing systems. 2013.