2020 IEEE Region 10 Symposium (TENSYMP), 5-7 June 2020, Dhaka, Bangladesh

Multiclass Classification for Bangla News Tags with Parallel CNN Using Word Level Data Augmentation

Ruhul Amin, Nabila Sabrin Sworna, Nahid Hossain
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh
ruhulamin6678@gmail.com, nsworna153069@bscse.uiu.ac.bd, nahid@cse.uiu.ac.bd

***Abstract*— Text mining is the procedure of exploring large unorganized text data. Due to the availability of numerous amounts of text data through online blogs, newspapers and other media, text classification and categorization is the hot topic nowadays. Many researches have been done on this topic on English and other western languages. However, very few notable researches have been on Bangla language. Unavailability of a notable dataset in Bangla language is another burden to develop a high-performance text classification tool. In this paper, we have presented a Bangla news tags classification approach. The classification has been done entirely based on news titles only with parallel Convolutional Neural Network (CNN) which is a category of deep neural networks utilizing word-level data augmentation approach. Due to the unavailability of a proper and updated dataset on Bangla news titles and tags, we have developed our own dataset which consists of 88,968 news titles and tags by scrapping online newspapers. According to the classification result, our approach shows an accuracy of 93.47% which is the highest amongst the similar works.**

Keywords—Bangla, News, Tags, CNN, Data Augmentation

I. INTRODUCTION

In the 21st century, data are the most valuable resource and even consider as the fuel of the century [1]. Text data are now available in several sources such as online blogs, newspapers, books and all other digital media. However, most of these data are unstructured and needs to be converted into structured useful information. From the last decade, newspapers are the common source of text data and most of those are unstructured. For easier access to users' interest and efficient use, it is necessary to categorize news so that it could easily be accessed. Bangla (endonym Bengali) is the 7th most spoken language in the world with around 228 million total speakers worldwide [2]. A large number of newspapers in both Bangladesh and India are publishing thousands of news articles in Bangla every day. Due to the increasing number of Bengali news readers, it is highly necessary to organize Bangla news articles in different tags. Bangla is a highly inflected language with a complex structure of sentences and grammatical rules [3]. Thus, solving a classification problem in Bangla is a challenging task. Due to lack of efficient news tags classifier and dataset in Bangla language motivate us to build an efficient Bangla news tags classifier and dataset.

As we mentioned earlier, very few notable researches have been done on Bangla language. In 2014, Abu et al. projected Bangla news classification using naive Bayes classifier [4]. Their approach is very straight-forward. At first, they removed the stop words from the article and then applied Naive Bayes classifier. In the same year, Ashish et al. suggested an approach for web document categorization [5]. They have explored four different supervised machine learning approach and Support Vector Machine (SVM) shows the highest accuracy of 89.14%. Quazi et al. presented a modified approach for text categorization from which they get 92.79% accuracy in 2018 [6]. They used TF-IDF with TF threshold for feature extraction. They also applied SVM as their classification method. In the same year, Tanvir et al. projected a Bangla article classification approach [7]. They have also developed a large dataset of articles. They apply different supervised learning models to compare their performances. They also applied Word2Vec, TF-IDF for feature extraction. Ankita et al. in 2018, proposed that the addition of Inverse Class Frequency (ICF) with Term Frequency (TF) and Inverse Document Frequency (IDF) gives the better feature extraction technique from Bangla [8]. They combined these three approaches for feature extraction then fed this into Multi-Layer Perceptron (MLP) to train the model. In order to get a better understanding, we have studied the following approaches in English languages as well in [9][10][11][12].

In this paper, we have illustrated a comprehensive multiclass classification approach of Bangla news tags. We have used parallel 1D CNN for classification utilizing word-level data augmentation technique. We preprocess our data before applying CNN by removing punctuations and stop words, tokenizing and applying word embedding. To produce word embeddings, we have used Word2vec technique. Due to lack of large and updated dataset on news titles and tags, we have developed a spider using Python's advanced Beautiful Soap [13] library to scrape our own dataset. We have collected data from different online sources. Since our approach is to categorize news based on titles, our dataset contains only the title of news and its corresponding tag. We have covered seven different tags and some tags have fewer samples. Thus, we have used oversampling in order to achieve better performance.

The paper is arranged as follows: In section II, we explained the proposed system and described our work and algorithm in detail. The paper illustrates experimental results and performance evaluation in section III, while we conclude the paper mentioning the future works and limitations of our system in section IV.

II. PROPOSED METHOD

In this section, we have illustrated the outline and development methodology of our proposed system with appropriate examples, figures and tables. The structure of the whole system has been shown in Figure 1.

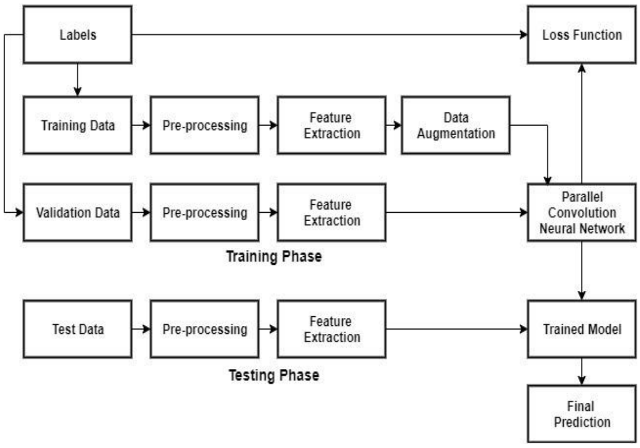
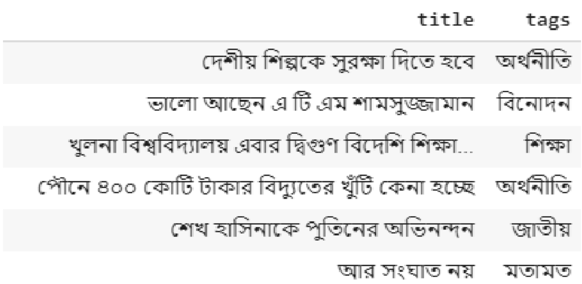


Fig. 1. Structure of the proposed system

A. Experimental Dataset

As we have mentioned earlier, due to unavailability of a robust, large and up-to-date dataset in Bangla, we have developed our own dataset. In order to collect data automatically, we have used Beautiful Soap [13] to develop a spider. The spider automatically crawls and scrapes different sources to collect titles and tags. The spider has been designed to collect only titles and tags not news body since our method classify tags based on titles, not news body. We have collected titles for the following 7 tags: বিনোদন(Entertainment), জাতীয় (National), আন্তর্জাতিক (International), মতামত(View Point), খেলা (Sports), অর্থনীতি (Economy), শিক্ষা (Education). In order to make the dataset robust, we have collected data from 10 different Bangla online newspapers. They are প্রথম আলো (Prothom Alo), ইত্তেফাক (Ittefaq), সমকাল (Samakal), কালের কন্ঠ (Kaler Kantho), ইনকিলাব (Inqilab), বাংলাদেশ প্রতিদিন (Bangladesh Pratidin), যুগান্তর (Jugantor), মানবকন্ঠ (Manobkantha), বাংলা ট্রিবিউন (Bangla Tribune), বাংলানিউজ২৪ (Banglanews24). Since our classification is based on news titles only, we have scrapped and stored titles and corresponding tags from the news articles (shown in Figure 2).



title	tags
দেশীয় শিল্পকে সুরক্ষা দিতে হবে	অর্থনীতি
ভালো আছেন এ টি এম শামসুজ্জামান	বিনোদন
খুলনা বিশ্ববিদ্যালয় এবার দ্বিগুণ বিদেশি শিক্ষা...	শিক্ষা
পৌনে ৪০০ কোটি টাকার বিদ্যুতের খুঁটি কেনা হচ্ছে	অর্থনীতি
শেখ হাসিনাকে পুতিনের অভিনন্দন	জাতীয়
আর সংঘাত নয়	মতামত

Fig. 2. Snippet of our dataset

Our dataset has in total 88,968 news titles. We have shown the title count of different tags in Table 1.

Table 1. Size of different tags of our corpus

Tags	Train	Test
বিনোদন(Entertainment)	9723	2374
জাতীয় (National)	14895	3766
আন্তর্জাতিক (International)	11632	2553
মতামত (View Point)	10546	2475
খেলা (Sports)	13133	3272
অর্থনীতি (Economy)	6445	1709
শিক্ষা (Education)	5028	1417

B. Data Preprocessing

Data preprocessing is one of the most significant subtasks for any text classification problem [14]. This step is pivotal in influencing the quality of the classification stage. Here are the steps that have been followed for data preprocessing.

- *Punctuation Removal:* Bangla textual data contains many punctuation characters which have less significance in the classification. So, we have removed some special symbols (<, >, :, ;, |, (,), ,, [,], ,! etc.) from the corpus.

  For example, রাজধানী (জাতীয়) converted to রাজধানী জাতীয় and বিয়েতে 'অনাকাঙ্ক্ষিত' অতিথি ট্রাম্প converted to বিয়েতে অনাকাঙ্ক্ষিত অতিথি ট্রাম্প.

- *Tokenization:* Tokenization is the procedure of detaching the text into useful tokens or words delimited by white space, punctuation or newline etc. We have used python string split function for tokenizing the text.

- *Stop Words Removal:* Stop words are those words that frequently appear in the corpus but do not integrate satisfactory information [15]. So, we need to remove these words before feature extraction. In English these types of words are: the, and, on, to, for etc. These words have no significance in the classification. Similar words in Bangla are এবং (and), ইহা (it), অনেক (many), তারপর (then) etc. We have removed these words in the preprocessing step. It helps to lessen the number of feature's dimensions. We have combined stop words from different sources [16][17][18] and created our Bangla Stop Words dataset by removing redundant words.

C. Feature Extraction

Text itself cannot be fed to deep learning models as they require some sort of numeric characterization of text. There are various types of approaches that can be taken for converting text into meaningful numeric representations. We have used word2vec technique for constructing word embeddings (vector representations for words) from the corpus.

- *Word Embedding:* Word2vec is extensively used in many tasks of natural language processing. Word2vec can preserve semantic and syntactic patterns of words throughout the corpus [17]. It is a shallow neural network that can regenerate morphological contexts of words. Word2vec receives text data as input and constructs a vector space, conventionally of several hundred dimensions for each word. Each distinctive word has the corresponding vector allocated in the space. These vector represented words are located in the space such that words sharing similar state of affairs in the corpus are positioned nearer to each other. Before training word2vec model, we replace words whose frequency is less than 5 with <UNK> which refers to 'unknown'. We have also replaced English or any foreign words with <ENG> and numbers with <NUM> tags. For constructing word2vec representations from our corpus, python gensim package has been utilized. We have used continuous skip-gram architecture with a context window of size 8. In our word2vec model, each word is represented with a vector size of 100. In the corpus, the maximum word count of the instance is 18. We have applied post zero padding to the instances having a length less than 18.

- *Oversampling:* Most standard deep learning algorithms encounter a significant drawback on performance with the imbalanced class distribution. Our dataset has skewed data distribution between classes. There are several popular techniques such as Synthetic Minority Over-Sampling Technique (SMOTE) that have been developed to deal with class imbalance problems. But it does not work well on high dimensional data [20]. We also find it from the experiment that it does not improve overall performances. After vectorization, each instance of our dataset converts into 18x100 length vector which makes it high dimensional. So we have augmented our dataset in the following way. In word2vec representations, similar words remain nearby in the hyperplane. First, we select an instance of the minority class from a uniform distribution with replacement. Then for each word in that selected instance, we find its nearest words from our word2vec vocabulary using K-Nearest Neighbors (kNN) algorithm. We have used the value of K=3 in our experiment. From those neighbors, we randomly select a word and for the rest of the words, we repeat the process to generate a synthetic instance. These augmented instances don't make any meaningful sentences but the underlying vector representations are quite close to the real instances. This facilitates the convolution layers to learn more position invariant local features thus assisting the fully connected layers to learn more robust decision boundary.

D. Methodology

Since we are interested only in local features, we have used 1D Parallel CNN for our classification model. Another efficient option is Recurrent Neural Network (RNN). RNN works well when dealing with sequential information. However, in our text classification, we don't need sequential information rather we need local features. In addition, RNN is also conflicting with our data augmentation technique [21]. Thus, we have used CNN for the classification model.

E. Model Architecture

CNN is evidently good at extracting position invariant features [21]. We have two parallel branches of CNN to extract local features for our model. The vector representations of each instance pass simultaneously through the 1D CNN branches. The model architecture has been shown in Figure 3. Both of the branches work as an individual feature extractor for the classifier. This allows an instance to be dealt with different resolutions or different n-grams (groups of words) autonomously, whilst the model comprehends how to best integrate these interpretations of contextual information [22]. In our model, the leftmost branch extracts features by bigram approach and the rightmost branch by trigram approach. Both of these branches learn important distinguishing features from local contextual information. Each of these two branches independently produce a matrix whose dimension is mi x ni. Then, we flatten these two matrices and combine them as a one-dimensional vector and pass it to the densely connected layers so that all the important distinguishing features can be learnt properly. In the intermediate layer, we have applied the ReLU(Rectified Linear Unit) activation function. In the final layer, we have used the Softmax activation function. To prevent overfitting and make the model generalized to out of sample data, we use a dropout rate of 0.4 and early stopping mechanism.

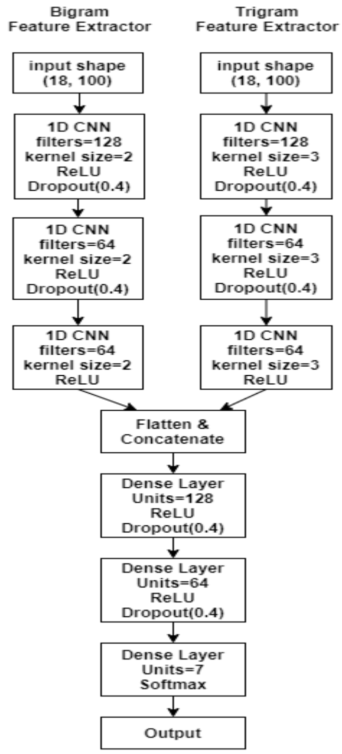


Fig. 3. Model architecture

We have used Categorical Cross Entropy as loss function and have used Adam optimizer with a learning rate of 0.001.

Pooling layers are convenient to shrink the dimensionality of output feature maps. It enables a model to be computationally less expensive. But pooling operations lose spatial information as well as local order of words. Moreover, in our experiments pooling layers have shown to degrade the performance of the classifier. Thus, we opt out of using any pooling layers in our model. The hyperparameters that have been tuned for our model are as follows:

- *Layer:* kernel size and number of filters for each convolution layer, number of units in each fully connected layer.
- *Function:* optimizer, activation function, loss function.
- *Rate:* learning rate, decay rate, dropout rate.

In order to tune our model's hyperparameters, 5-fold cross validation has been utilized so that we can achieve the best validation results. The values of the finally picked hyperparameters have been shown in Figure 3.

III. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we have illustrated the experimental results and performance evaluation. The experiment has been performed on Kaggle Nootbook. Kaggle Nootbook is a free cloud computational service owned by Google Inc. It provides 5 GB disk space, 13 GB RAM and NVIDIA Tesla P100 GPU with 16 GB memory [23]. We have used python deep learning framework Keras with TensorFlow backend to develop our model which comes as pre-installed with Kaggle Notebooks. The average K-Fold score of our model has been shown in Table 2.

Table 2. Summary of average K-Fold Score

Model	**Average K-Fold Score**
Parallel CNN	92.65

In this experiment, we have used four performance evaluation metrics and they are Accuracy, Sensitivity or Recall, Precision, and F1 Score. These metrics can be defined as follows:

Accuracy = (TN + TP)/(TN+TP+FN+FP)…………….(1)
Precision = TP/(TP + FP)……………………………...(2)
Recall = TP/(TP +FN)……………………………...(3)
F1 Score =2*((Precision*Recall)/(Precision+Recall))..(4)
Where:
TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

The result of the experiment has been shown in Table 3.

Table 3. Summary of results

Model	Parallel CNN
Accuracy	93.47
Recall (weighted avg.)	90.07
Precision (weighted avg.)	91.62
F1-Score (weighted avg.)	91.35

Our model shows significantly high accuracy of 93.47 per cent. However, recall of the model is 90.07 per cent and thus, requires improvement.

IV. CONCLUSION AND FUTURE WORK

Text classification in Bangla is a difficult task. In our work, we have classified article from the titles only not observing the whole article and it is very challenging. We have used 1D CNN which has 2 parallel branches to extract features. We have demonstrated all the procedure and techniques we used during the project in this paper. We have achieved significantly higher accuracy and the highest amongst similar works. We have built a large dataset of latest news titles and tags, also we have compiled a dataset of Bangla stop words. These datasets will be freely available in our GitHub Repository for other researchers. The key challenge that we have faced is improving the recall and precision of the classifier. Our data augmentation technique improves the performance to a certain extent but there is still scope for further improvements. In future authors would like to use Generative Adversarial Network (GAN) to create artificial instances and improve the overall performances of the classification task.

REFERENCES

[1] "The World's Most Valuable Resource Is No Longer Oil, but Data." The Economist, The Economist Newspaper, www.economist.com/leaders/2017/05/06/the-worlds-most-valuableresource-is-no-longer-oil-but-data.

[2] Ethnologue. (2019). Summary by language size. [online] Available at: https://www.ethnologue.com/statistics/size .

[3] Nayan Banik, Md. Hasan Hafizur Rahman, (2019) "Evaluation of Naive Bayes and Support Vector Machines on Bangla Textual Movie Reviews", Intelligent Computing-Proceedings of the Computing Conference. Volume: 998, Pages: 93--98, Publisher: Springer.

[4] Chy, Abu Nowshed & Seddiqui, Hanif & Das, Sowmitra. (2014). Bangla news classification using naive Bayes classifier. 16th Int'l Conf. Computer and Information Technology, ICCIT 2013. 10.1109/ICCITechn.2014.6997369.

[5] Ashis Kumar Mandal, Rikta Sen, (2014) "Supervised learning methods for bangla web document categorization", arXiv preprint arXiv:1410.2045, 8 October.

[6] Quazi Ishtiaque Mahmud, Noymul Islam Chowdhury, Md Masum, (2018) "Reducing Feature Space and Analyzing Effects of Using Non Linear Kernels in SVM for Bangla News Categorization",2018 International Conference on Bangla Speech and Language Processing (ICBSLP). Pages: 1--6, Publisher: IEEE.

[7] Md Tanvir Alam, Md Mofjul Islam, (2018) "BARD: Bangla Article Classification Using a New Comprehensive Dataset",2018 International Conference on Bangla Speech and Language Processing (ICBSLP). Pages: 1--5, Publisher: IEEE.

[8] Ankita Dhar, Niladri Sekhar Dash, and Kaushik Roy, (2018) "Categorization of Bangla Web Text Documents Based on TF-IDFICF Text Analysis Scheme",Annual Convention of the Computer Society of India. Pages: 477--484, Publisher: Springer.

[9] TAISHI SAITO, OSAMU UCHIDA, (2018) "Automatic labeling to classify news articles based on paragraph vector", International Journal of Computers. Volume: 3,Publisher: International Association of Research and Science.

[10] Eric S. Tellez, Daniela Moctezuma, Sabino Miranda-Jime nez, Mario Graff, (2018) "An automated text categorization framework based on hyperparameter optimization",Knowledge-Based Systems, 1 June. Volume: 149, Pages: 110--123, Publisher: Elsevier.

[11] Boyi Yang, Adam Wright, (2018) "Development of deep learning algorithms to categorize free-text notes pertaining to diabetes: convolution neural networks achieve higher accuracy than support vector machines",arXiv preprint arXiv:1809.05814. Volume: 6, Pages: 2--6.

[12] Shadi Diab, (2019) "Optimizing Stochastic Gradient Descent in Text Classification Based on Fine-Tuning Hyper-Parameters Approach. A Case Study on Automatic Classification of Global Terrorist Attacks", arXiv preprint arXiv:1902.06542. Volume: 16, Number:12.

[13] "Beautiful Soup Documentation." Beautiful Soup Documentation Beautiful Soup 4.4.0 Documentation, www.crummy.com/software/BeautifulSoup/bs4/doc/.

[14] V. Srividhya, R. Anitha 2010, 'Evaluating Preprocessing Techniques in Text Categorization', International Journal of Computer Science and Application, ISSN 0974-0767

[15] A. Rajaraman. J.D. Ullman, "Data Mining". Mining of Massive Datasets, Cambridge, England, CUP, 2011, ch 1, pp. 1-17.

[16] https://www.ranks.nl/stopwords/bengali

[17] https://github.com/stopwords-iso/stopwords-bn

[18] https://sanjir.com/6202/

[19] Mikolov, Tomas; et al. (2013). "Efficient Estimation of Word Representations in Vector Space". arXiv:1301.3781

[20] Rok Blagus, Lara Lusa, (2012) "Evaluation of smote for highdimensional class-imbalanced microarray data",2012 11th International Conference on Machine Learning and Applications. Volume: 2, Pages: 89--94, Publisher: IEEE.

[21] Yin, Wenpeng, et al. "Comparative Study of CNN and RNN for Natural Language Processing." ArXiv:1702.01923 [Cs], Feb. 2017. arXiv.org, http://arxiv.org/abs/1702.01923.

[22] Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." ArXiv:1408.5882 [Cs], Sept. 2014. arXiv.org, http://arxiv.org/abs/1408.5882.

[23] "Notebooks Documentation." Kaggle, www.kaggle.com/docs/kernels ?fbclid=IwAR0a4pxbvpQetZ6NO8rvsLXQ9SOzcqCJG1a4DP8N-Dy-ZzOfxrFDwdYkwrM#technical-specifications.