Journal of the Institute of Engineering
Volume 17, No. 1 Published: April 2023
ISSN: 1810-3383

Political Profiling of Nepali Twitter Users using Vector Model

Arun K. Timalsina [a], Ramesh Kharbuja [b]

[a]*Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Kathmandu, Nepal*

[b]*Institute of Science and Technology, Bhaktapur Multiple Campus, Tribhuvan University, Bhaktapur, Nepal*

Corresponding Author: [a]t.arun@ioe.edu.np,
[b]ramesh.kharbuja1@bkmc.tu.edu.np

Received: 2023-03-18 Revised: 2023-04-06 Accepted: 2023-04-07

Abstract:
Everyday people in social networks create a huge amount of data as posts, blogs, tweets, articles, comments, etc. in the form of text, images, audios and videos. The number of social media users and the data they are adding up in cloud is increasing drastically day by day. People from all over the globe with different region, culture, language, education, public figures posts or blogs reflecting their vision and opinion. These micro-blogs are now being used by researchers and business houses for assessing customer opinion to their implicit intension and behavior. Using the tweet contents, this research is to classify a Nepali twitter user to one of the pre-defined class of political parties in Nepal using vector space model. In this approach a set of words is defined as document class that represents to a political party. A number of steps for text-preprocessing is to be done based on morphological structure of Nepali language for the better result. TF-IDF and Doc2Vec methods are used to extract the feature of the terms being used in tweets. Similarity measure is used to match the tweeter's profile with political party's class through similarity matching score. Vector model-based TF-IDF and Doc2Vec methods are compared for their effectiveness in the domain of tweets in Nepali language.

Keywords: Nepali language, Political profiling, Vector model, TF-IDF, Word-embedding, Word2Vec, Doc2Vec, Cosine similarity

1.Introduction

One of the most popular online activities of people is the use of social media like Facebook and Twitter. As per the global statistics [1], over 4.26 billion people were using social media worldwide in 2021, which is projected to increase to almost 6 billion by 2027. The growth of people engagement in social media is unprecedented. There is no barrier of region, culture, language, and education level of people for making their posts on social media. Similarly, there is huge diversity on their expression on topics like lifestyle, business, politics, weather, sports, travel, violence, etc.

Social network users from different cultures and backgrounds regularly post and tweet large numbers of textual arguments reflecting their opinion and perspectives in different aspect of life and make them available to everyone. In this context using social media for political discourse is becoming common practice, especially around election time. Prediction of the public opinion about the election campaign and results are becoming the major interest for many researchers and the press media. There was an intense competition between Donald Trump, representing the Republican Party, and Hillary Clinton, representing the Democratic Party in the 2016 American Presidential Election [2]. During the election, there were various discussions and heavy postings regarding Hillary and Trump among the users of online social networks like Twitter.

A rich document might be composed of texts, music, and images, however, text is referred in most of the cases. It involves in automating the classification of documents to different clusters regarding attitudes, opinions, emotions, ethnicities, religions, etc. It involves introducing the degree of positivity,

negativity or neutrality pointed in document towards the pre-defined classes. Sentiment analysis is one of its applications. Many text analysis related works are done in English language, which involves less and simple text pre-processing tasks as compared to morphological rich languages. Nepali is one of the morphological rich language in which Devanagari script is being used.

Most of the text classification related works have been done in English language. Languages like Nepali is morphologically rich that means more complex in structure in formation of words. Nepali is a high inflectional language. A single word has more than one affix, such that it may be expressed as a combination of prefix and suffix. Nepali has some variants in spelling and typographic forms mostly while using in informal writings such as in social media, personal conversations and messaging.

As there is very limited research on political disclosure of tweeters using Nepali language, the research goal of this work is to generate political profiling of Nepali twitter users based upon their tweet contents on social media. Predicting the degree of bias on the expression of a person is in fact, finding the true profile and commitment of such political person towards his/her political affiliation. Comparative study using two different vector methods; TF-IDF and Doc2Vec, had been accomplished with evaluation measures using accuracy and F1-scores.

The research data in the form of corpus is being collected and processed from twitter dataset. Selective tweets from Nepali political leaders, ex-secretaries of Nepal Government, social activists and some renowned journalists were collected through web scraping which were published on Twitter up to December 11, 2019. These key figure selection is also based upon their high-frequency of twitter posting public records on their own political linkages and other categories.

2. Literature Review

Text classification is done for the categorization of text (including sentence, paragraph, and complete articles) according to the combination of these words and the context of that text. In Natural Language Processing (NLP), text classification is a primary tasks with broad applications such as Sentiment Analysis, Opinion mining, Text Summarization, etc.

Unstructured raw data is generated in huge amount in every aspects of communication in the form of text such as emails, web pages and social media. Human is capable to perceive and process unstructured text data efficiently which is complex for machines to do the same. These are the primary source of information, but extracting the concerned data from these raw sources is challenging and time-consuming as they are not in structured form. Today, text classification is done by enterprises to enhance decision-making and automating various business processes.

Text classification can be broadly categorized into two different ways: first manual classification and automatic classification. Mixing these two techniques a hybrid way can be built up as well. In manual way, a person is responsible for understanding the context of text and categorizes it accordingly. The second one deploys NLP based machine learning for building up a model. Nevertheless, it may be time consuming at the first to train a machine, later it can automatically classify text in a faster and more cost-effective way as compared to manual one. Combining these two, a hybrid model can also be derived for complex data.

A. Related Works

Caetano et al. [3] analyzed the political homophily among Twitter users during the 2016 American Presidential Election from 4.9 million tweets of 18,450 users and their contacts. Users were classified into classes with Trump supporter, Hillary supporter, positive, neutral, and negative regarding their sentiment towards Donald Trump and Hillary Clinton. Secondly, political homophily in different scenarios were analyzed.

Similar research on Irish general election of 2011, Bermingham and Smeaton [4] predicted electoral outcome. Supervised classification with unigram features was used to analyze political sentiment in tweets achieving 65% accuracy on the task of positive, negative, and neutral classification. Overall volume turned out to be a stronger indicator than the perceived opinion through sentiment analysis on

such election outcomes.

Language specific research using TF-IDF with Support Vector Machine (SVM) and Naïve Bayesian (NB) for developing document vector had been done in [5, 6]. TF-IDF based implementation efficiency measures like accuracy and time to get the result for each classifier and determined the accuracy of Arabic text classification [5] and Nepali spam filtering on Nepali language based short messages [6] are compared accordingly.

Dangol and Timalsina [7] implemented various Nepali morphology specific features such as removing stop-words, removal of word suffices using Nepali language morphology to reduce the number of dimensions in Vector Space Model.

Similarly, Kafle et al. [8] classified documents using word2vec and simplifies the process of automatically categorizing Nepali documents while increasing the precision and recall. They compared three techniques SVM with TF-IDF, cosine similarity with TF-IDF and SVM with Word2Vec and concluded that the SVM with Word2Vec model outperforms the remaining.

Radu et al. [9] compared TF-IDF with Doc2Vec model for embedding by combining separately with distance measuring algorithm like K-Means, Spherical K-Means, Density Based Algorithm (DBSCAN) and Topic Modeling Algorithm like Latent Dirichlet Allocation (LDA). The finding was Doc2Vec beats TF-IDF with more accuracy and precision and other performance metrics.

B. Related Theory

Feature Selection and Extraction

Human brain is highly sensitive to pictures, graphs, and sound but computer or machine processes numbers for computation. NLP and machine learning algorithms generally plays with numeric data. So transforming text into numbers is the primary task in this field which is known as Text Vectorization or feature extraction. Extracting information in the form of features from the text data is in fact the technique which is to reduce the dimension and identify the important features.

Bag of words

Bag of words deals with the frequency of words in document and is the simplest feature extraction method. It gives the word-features dictionary from all of these words taken into consideration. It is known as a “bag” of words, since the method doesn't care about the order of the word, it only checks if the word occurs or not in a group of words.

TF-IDF

One of the drawbacks of bag of words method is that the words with higher frequency becomes dominant in the document and there is chance of ignoring such words having more significance to domain but with less frequency of occurrence in the document. Term Frequency – Inverse Document Frequency (TF_IDF) scales down the score of these words that are frequently common in all documents. It generates the meaningful score of the words that are unique which gives the significant meaning and importance in a particular class. It has been used by Google as a ranking factor for the web content for a long time.

For a word *w* in a document *d*, IDF of word '*w*' is given by:

$$\text{IDF}w = \log(\frac{\text{N}}{\text{DFw}})$$

And final TF-IDF of a word *w* is given by:

$$TF\_\text{IDFw} = TF\text{w} * \log(\frac{\text{N}}{DF\text{w}})$$

Where,

TF: number of occurrences of *w* in document *d*.
DF: number of documents containing the word *w*.
N : total number of documents in the corpus

Word2Vec

Word2Vec [10] is a word embedding technique widely used for text extracting features for text classification. It is a two-layer neural networks, which builds a semantic context of words. The model takes a huge corpus of text as an input and gives the multi-dimensional embedding of words. Words having common contexts are placed in near proximity in vector space. Word2vec is used with generally two architectures: skip gram or continuous bag of words. Targeted word is used to predict the neighboring words using skip gram architecture

whereas targeted word is predicted using the surrounding words in continuous bag of word architecture. Word2Vec is a two-layer neural networks taking a large corpus as input and producing a vector space with multiple dimensions. Algorithmically, continuous bag of word and skip gram both models are similar.

Doc2Vec/Paragraph Vector

Doc2vec also known as Paragraph Vector [11] is a vector which represents the documents built using the vectors contained in the documents by using some averaging tool. It does not depend upon the length of the document which means it is applicable to sentences, paragraphs, and documents of any length.

Cosine Similarity as Classifier

Measuring the similarity between document classes is the main task in the text classification. The feature values extracted from the TF-IDF is single per term whereas the dimensionality of a word in Word2Vec is up to 300 dimensions and hence also for document in Doc2Vec. The proposed model uses cosine similarity measure with TF-IDF calculation or Doc2Vec for computing the similarity between two document classes of tweets relating to tweets with different political parties.

$$\text{Similarity}\,(\text{docA}, \text{docB}) = \frac{\text{A}.\,\text{B}}{|\text{A}| * |\text{B}|}$$

Similarity value ranges between -1 and 1 where -1 means completely dissimilar and 1 means completely similar to each other.

3. Methodology

A. Proposed Model

The proposed model consists of number of steps such as pre-processing and feature selection, feature extraction through TF-IDF, Doc2Vec using Word2Vec and classification using Cosine Similarity. Cosine similarity is used for comparing the similarity of documents using the result obtained from TF-IDF or Doc2vec.

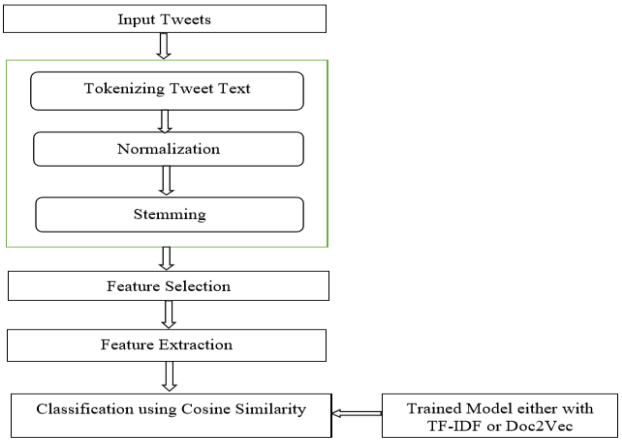

Figure 1: Proposed model of the system

B. Data Collection and Distribution

Tweets of political leaders, journalists, activists, ex-secretaries of Nepal governments are downloaded from twitter through the Tweepy tool which makes call to APIs provided by twitter. Total of 72 selected peoples' tweets with number of 254414 tweets of size 33.2 MB is scrapped from twitter. These 72 public figure selection was as per their social presence and or political status and threshold counts of tweets that they have posted on Twitter.

Table 1: Political class categorization

Twitters' Affiliation Classes		
नेपाली काँग्रेस	Nepali Congress	NC
नेकपा	Nekapa	NCP
मधेसवादी	Madhesbadi	MB
नया पार्टी	New Parties	NP
राप्रपा	Raprapa	RPP
गैर राजनीतिक	Non-Political	GR

The data consists of all these 72 people tweets up to 11-Dec-2019. After certain pre-processing, manual categorization of 94195 tweets of size 19.3 MB, has been done to introduce 6 different categories according to political affiliation and corpus of tweets as listed on Table 1.

Tweets of top level political leader are taken as the training and testing data for respective class of political parties. Some renowned personalities such as journalist, activist, and authors are considered as ‘non-political’ category during the feature extraction phase in order to improve accuracy of the model. Though there are less number of tweeters from Terai region, they are kept under the 'Madhesbadi party'. Tweets of Sajha Party, Bibekshil Nepali are

considered as New Parties as they are considered to be the alternative of old parties by youths all over the country. The diverse nature of the collected data is shown in Table 2.

Table 2: Tweeter and tweet data distribution

Category Class	Total Tweeters	Total Tweet Count	Average Tweet Count	Tweeters > Avg. Tweet
NC	13	11992	922	7
NCP	25	24437	977	12
MB	6	8642	1440	4
NP	14	24640	1760	9
RPP	5	9192	1838	4
GR	9	15292	1699	5
Total	72	94195	8636	41

C. Preprocessing and feature selection

The data preprocessing is the entry point process in text classification. In the proposed work only the Devanagari text/characters are taken. These Devanagari text are separated from other text using the Devanagari character code table which is "The Unicode Standard, Version 12.1" [12]. All the characters except punctuations, digits are taken into consideration. As the study is focused to Nepali language tweets, Nepali language based few morphological tasks are done including rhaso-dirgha, ekar and ukar, स and श ,ब and व, etc. Similarly, removal of tense, adjective, plural, gender related suffixes from the words are also done. Finally the stop words are also removed. Stop words are those language specific words which do not carry the significant meaning both semantically and contextually.

Feature extraction calculates features of document/word on the basis of frequency, order or context of words. The feature extraction is the process of representing document/words in such a way that facilitates the decision making for classification. Basically features are used as input for the classifier that assigns them to the class that they represent.

TF-IDF: Using TF-IDF all words of particular party and test individual tweets text are embedded to a single value numerical value. Doc2Vec: Word2Vec represents words of a particular party and test individual to multi-dimensional numerical values. Computing the average of the all the words in the corpus, document vector which represent the overall category i.e. party tweets can be generated.

The main aim of automated classification is to learn from training and make efficient decision to predict to which category the given input text lies on. Cosine similarity to measure the similarity between the party specific category and test individual tweets using the result obtained by feature extraction step is used.

D. Algorithm

The proposed model has following steps:

1. Generate tokens (term) for each political party defined from the tweets in profile of an individual that belongs to a political party that have been defined or concerned with. In this step all the Non-Devnagari words/symbols are removed. Generation of token involves splitting of words from the sentences of tweets. Taking a tweet of Congress Leader Gagan Thapa as examples. "उप-चुनावले नेकपालाई स्पष्ट सन्देश दिएको छ-सरकारको शैली सच्याउ नत्र सकिन्छौ सच्चिने वा सक्किने उसको कुरा!". Generated Tokens are "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा","!".
2. Replace *Murdhanya* (ट, ठ, ड, ढ, ण) to *Dantya* (त, थ, द, ध, न) ,श to स , all Rhasyawo Ekar, Ukar to Dirgha. Remove Purnabiram, Halanta. The tokens changed to "उपचुनावले", "नेकपालाई", "स्पस्त", "सन्देस", "दीएको", "छ", "सरकारको", "सैली", "सच्याऊ", "नत्र", "सकीन्छौ", "सच्चीने", "वा", "सक्कीने", "उसको", "कूरा","!".
3. Suffixes words are removed such as एको, एका, एकी, ले, लाई ,बाट ,देखि. The tokens converted to "उपचुनाव", "नेकपा", "स्पस्त", "सन्देस", "दीए", "छ", "सरकार", "सैली", "सच्याऊ", "नत्र", "सकीन्छौ", "सच्चीने", "वा", "सक्कीने", "उस", "कूरा".
4. Remove stop words from the complete sets, stop words are collected from various words. They are also generated from tweets set with maximum frequency in entire corpus. The remaining tokens are "उपचुनाव", "नेकपा", "स्पस्त", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने".

5. Using the training documents, Calculate weight of remaining terms for each document class using one time with TF-IDF and another with Word2vec. This step is also called feature process. TF-IDF generated a single feature for each term as in Table 3 whereas Word2Vec generates several hundred normally (100-300) dimensional features for each term. As shown in Table 4, 300 dimension is used in proposed model.

Table 3: Weights of sample terms for different political classes extracted by TF-IDF model

		Term-Weight evaluated for different political classes			
		NC	NCP	MB	RPP
Terms	उपचुनाव	0.00000000256894	0.00000000000000	0.00000000000000	0.00000000000000
	नेकपा	0.00000000001200	0.00001598500000	0.00000000000000	0.00000000009850
	सन्देश	0.00000000125000	0.00000000000000	0.00000000000000	0.00000000036800
	सरकार	0.00000056852100	0.00002882100000	0.00000452180000	0.00000025487900

6. Compute the term-document weight i.e. document vector from above weights. In case of word2vec a single word is represented by 300 dimensional feature vector; Table 4. A document vector is the resultant of all the vectors that represented the words contained in the document. It can be assumed like the result of multiple vectors with different magnitude and directions.

Table 4: Sample Word2vec Feature weights of different classes

	Dimension1	Dimension2	Dimension3	.....	Dimension299	Dimension300
NC	0.14801884	0.17013658	0.09433010	.....	0.04762081	-0.32374550
NCP	0.14691746	0.28224322	0.01402679	.....	0.05902386	-0.21432162
MB	0.04258833	0.17742235	0.05189126	.....	-0.07656067	-0.29571226
NP	0.17958795	-0.26892216	-0.17468330	.....	0.11412652	-0.32581979
RPP	0.04626197	0.03916102	0.06258884	.....	0.09059965	-0.27342925
GR	0.09219553	0.18397406	0.04860130	.....	0.00784407	-0.12920160

Compute the document vector of a test documents following the above all steps. See Table 5 as tweeter containing the sample words "नेकपा", "सन्देस", "सरकार", "देश" and corresponding term-weight evaluation as preprocessing.

Table 5: TF-IDF weights of sample terms in test document

Terms	Term Weight
नेकपा	0.0000000000120
सन्देस	0.0000000012500
सरकार	0.0000005685210
देश	0.0000000001210

Table 6: Sample Doc2Vec feature association in test document

Tweet Dimensions	Dimens-ion1	Dimens-ion2	Dimens-ion3	.....	Dimens-ion299	Dimens-ion300
Person X	0.101884	0.1013658	0.433010	.....	0.062010	0.237500

7. Evaluate the similarity measures between the test document and the document class i.e. political party's document using cosine similarity. Table 6 depicts such association similarity.
8. Classify the test document to the political party that gives the maximum similarity measure. As in Table 7, a sample text of tweets from a Congress leader's timeline, cosine similarity provides the result and similar for others.

Table 7: Random NC Tweeter's tweet similarity using the TF-IDF

Political Class	Similarity
NC	0.581329119
NCP	0.538390974
MB	0.377371835
NP	0.446373845
RPP	0.553099221
GR	0.540828569

9. Repeat all the above processes to evaluate according to the K-fold cross validation method. In Phase 1 tweets of all 72 tweeters are used where as in Phase 2 only 41 tweeters' tweets are taken into consideration who tweets more than average number of tweets in their respective class. In each phase of experiment, K with value 10 and 5, K-fold cross validation is done. The tweeters in the test fold is tested one by one with model and found out maximum similarity measure with all political class as classified by cosine similarity and assigned to the class having maximum similarity measures.

4. Result and Evaluation

Two different methods, TF-IDF and Doc2Vec models based classification performances are compared using metrics like Accuracy, Precision, Recall and F1-Score. The results are verified through 5-fold and 10-fold cross validations.

Confusion matrix

A classifier predicts all tweets data instances of a test dataset as either positive or negative. Any of the four outcomes true positive (TP), true negative (TN), false positive (FP) and false negative (FN) are listed.

TP and TN are the correctly classified observations, whereas FN is incorrect negative prediction and FP is incorrect positive prediction. Based on the resulted outcomes performance metrics, using confusion matrix, can be calculated.

Training and Testing of data is done in 2 phases. One with all the data of 72 tweeters whereas second phase is done with taking only the tweets whose number of tweets are more than average from their respective class. Again in each phase using 10-fold and 5-fold cross validation has been performed. Table 8 depicts sample result of 10-fold cross validation experiment.

Table 8: Confusion Matrix of 6-classes
(Sample count of 10 fold cross validation)

		Actual Class					
		NC	NCP	MB	NP	RPP	GR
Predicted Class	NC	7	1	0	0	0	0
	NCP	1	22	0	0	0	2
	MB	0	0	5	0	0	0
	NP	5	1	1	14	0	5
	RPP	0	0	0	0	5	0
	GR	0	1	0	0	0	2

For 10-fold cross validation, TF-IDF and Word2vec models are trained first using 90 percent of tweets and 10 percent of Tweets are used as testing purpose following K-fold cross validation method. In second phase, 80 percent of tweets are trained and 20 percent are tested.

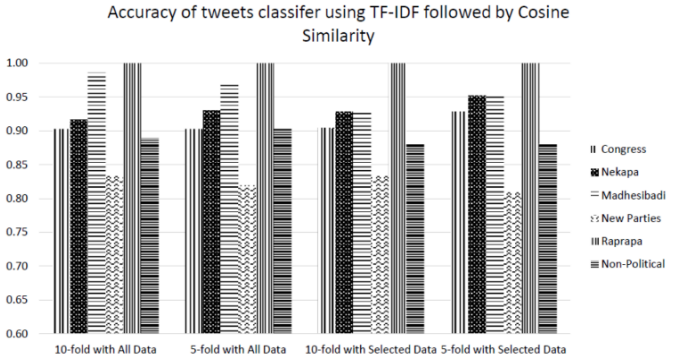

Figure 2: Accuracy scores with TF-IDF model

Accuracy and F1-score measures for both 5-fold and 10-fold experiments with TF-IDF model results are depicted on Figure 2 and 3.

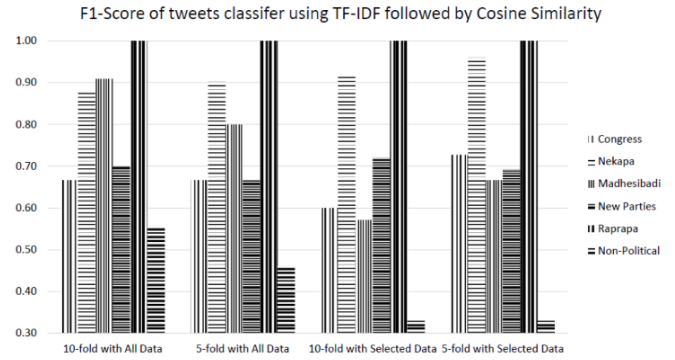

Figure 3: F1-score with TF-IDF model

Similarly, averages of all four measures accuracy, precision, recall and F1-score are visualized on Figure 4. Accuracy and F1-score measures for both 5-fold and 10-fold validation experiments with Word2Vec model results are depicted on Figure 5 and 6. Similarly, averages of all four measures are visualized on Figure 7.

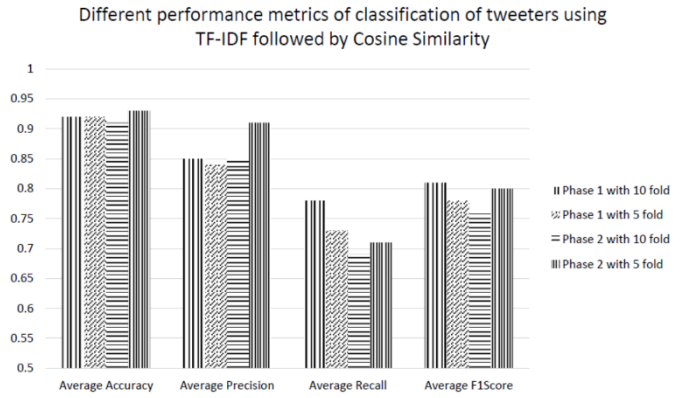

Figure 4: Performance-scores with TF-IDF model

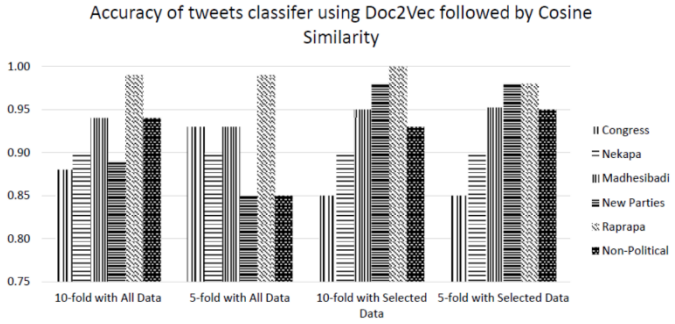

Figure 5: Accuracy with Doc2Vec model

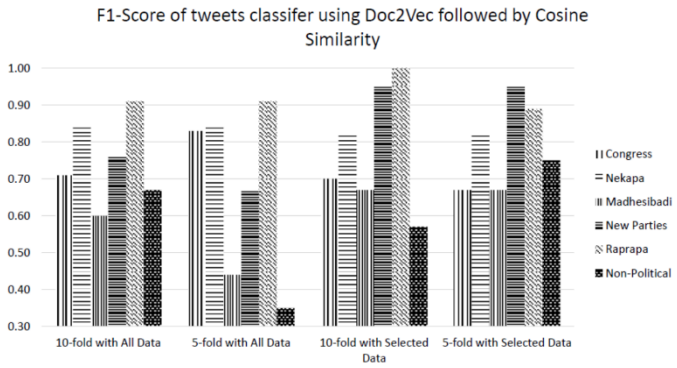

Figure 6: F1-score with Doc2Vec model

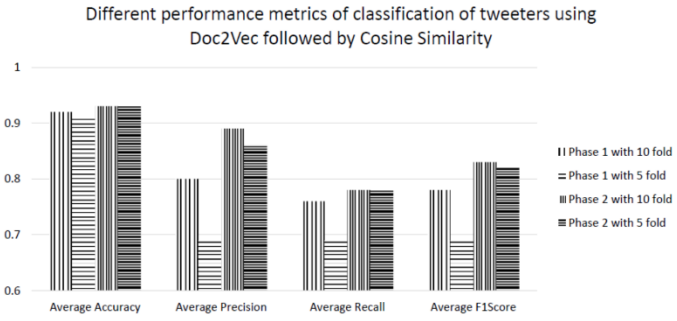

Figure 7: Performance-scores with Doc2Vec model

Overall two different vector method based performance comparison is depicted on Figure 8. Average accuracy of both methods reached up to 92%, however, the most robust measure, F1-score differed within 1 percent only on these two different methods. All of the measures turn out to be consistent, even with such sparse dataset considering the count of tweet terms, count of twitter users of each class, the number of tweet postings on social media by these political activists as well as non-political influential figures of Nepalese society.

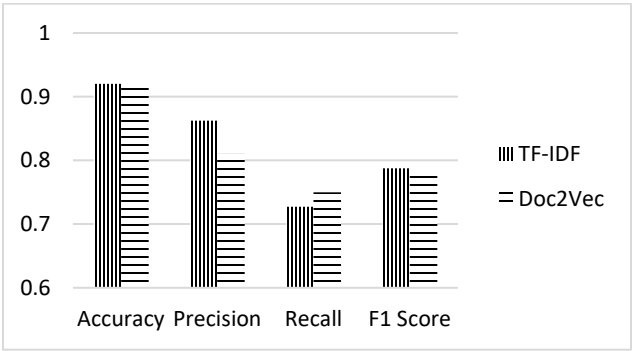

Figure 8: Performance comparison between TF-IDF and Doc2Vec

5. Conclusion

In this research work, tweets from twitter is used for training and testing data to perform a political profiling Nepali political leaders. A corpora for 6 political parties, using twitter data as a primary source, is being developed for the classification purpose. The research work compared between two feature extraction techniques TF-IDF and Doc2Vec models in Nepali tweets domain in which a tweeter is classified into one of the six classes: Nepali Congress, Nepal Communist Party, Madhesbadi Parties, New Parties, Raprapa Nepal and Non-Political category. Cosine similarity measure is used as classifier by computing the similarity of profile of tweeters' with pre-defined class of political parties.

In text classification and analysis works, features of the language is very important part to focus on. Number of steps in pre-processing of text depends on complexity in morphological structure or richness of the language. This work presents the classification of Nepali text represented in Devnagari script with Nepali language specific features which are different than English. There are huge repositories on English language based twitter data analysis, however, Nepali language based researches are very limited. In addition, political profiling of Nepali leaders based on their expression through tweet data in Nepali language is, probably, the distinctive research on this domain. Results with TF-IDF is slightly promising in comparison to Doc2Vec. One of the reasons of such performance of these experiments are due to the size of the dataset. It is found that in an average TF-IDF and Doc2Vec both achieved 92% accuracy. But in terms of F1-Score TF-IDF beats Doc2Vec with 79% and 78% respectively. A corpora of Nepali tweets is developed which can be made available for experimentation to other researchers and enthusiasts.

Future Enhancements

Result of the study shows that TF-IDF beats Doc2Vec method. In fact, Doc2Vec performs better than TF-IDF in other setups due to the reason that Doc2Vec also considers the context of terms while TF-IDF only considers frequency of terms in content. The performance augmentation is possible only through large amount of data with context, so more tweets can be collected from twitter in future and perform the analysis. Similarly, there is also possibility of experimentation with deep learning algorithms for achieving better result. Here only

some of Nepali language specific pre-processing are done, such as transforming the derived words to root words by removing the affixes, removal of stop words. A more sophisticated stemmer can be built by discovering more features in Nepali language which leads to better performance of classification done in this study.

References

[1] Number of social media users worldwide from 2017 to 2027 (in billions) https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/ (Accessed Feb 2023).

[2] Reuters. In breathless U.S. election, twitter generates buzz not cash. 2016. https://www.reuters.com/article/us-usa-election-twitter/in-breathless-us-election-twittergenerates-buzz-not-cash-idUSKCN12R2OV. (Accessed Feb 2023).

[3] Josemar A. Caetano, Hélder S. Lima, Mateus F. Santos and Humberto T. Marques Neto "Using sentiment analysis to define twitter political users' classes and their homophily during the 2016 American presidential election", Journal of Internet Services and Applications 9, 18 (2018). https://doi.org/10.1186/s13174-018-0089-0

[4] Adam Bermingham and Alan F. Smeaton. "Classifying sentiment in microblogs: Is brevity an advantage?" In CIKM '10: Proceedings of the 19th ACM international conference on Information and Knowledge Management (2010) Pages 1833–1836.

[5] Tarek Elghazaly, Amal Mahamoud and Hesham A. Hefny. "Political Sentiment Analysis Using Twitter Data." ICC '16: Proceedings of the International Conference on Internet of things and Cloud Computing, March 2016, Pages 1–5. https://doi.org/10.1145/2896387.2896396

[6] Tej Bahadur Shahi and Abhimanu Yadav. "Mobile sms spam filtering for nepali text using naive bayesian and support vector machine." International Journal of Intelligence Science Vol.4 No.1 (2014), Paper ID 40857, 5 pages.

[7] Dinesh Dangol and Arun K. Timalsina. "Effect of nepali language features on nepali news classification using vector space model." In Proceedings of the 10th International Conference on IT Application and Management (2013) Pages 132–138.

[8] Kaushal Kafle, Diwas Sharma, Aayush Subedi and Arun K. Timalsina "Improving Nepali Document Classification by Neural Network." In Proceedings of IOE Graduate Conference. (2016)

[9] Robert-George Radu, Iulia-Maria Radulescu, Ciprian-Octavian Truica, Elena-Simona Apostol and Mariana Mocanu. "Clustering Documents using the Document to Vector Model for Dimensionality Reduction." Conference Paper in Computer Science and Engineering Department, Faculty of Automatic Control and Computers University Politehnica of Bucharest, Bucharest, Romania. (2020)

[10] Tomas Mikolov, Ilya Sutskever and Kai Chen "Distributed Representations of Words and Phrases and their Compositionality", Research Work in Google Inc. (2013)

[11] Tomas Mikolov and Quoc Le. "Distributed Representations of Sentences and Documents." Proceedings of the 31st International Conference on Machine Learning, Beijing, China. (2014)

[12] Devanagari Range: 0900–097F http://www.unicode.org/charts/PDF/U0900.pdf Accessed 20 June 2019.

[13] Wahyu S. J. Saputra. "Calculate Similarity of Document using Cosine Similarity to Detect Plagiarism" Bali International Seminar on Science and Technology, Bali, Indonesia. (2011)

[14] Bal Krishna Bal and Prajol Shrestha "A Morphological Analyzer and a stemmer for Nepali." Published by Madan Puraskar Pustakalaya, Nepal. (2005)