Journal of Science and Technology
ISSN: 2362-1443 (Print) 3059-9490 (Online)
2024, Vol. 4, No. 1, 14-20

Political Profiling of Nepali Twitter Users based on Space Vector Model

Ramesh Kharbuja
Institute of Science and Technology,
Tribhuvan University
rankerramesh@gmail.com

Arun K. Timalsina
Institute of Engineering,
Tribhuvan University
t.arun@ioe.edu.np

Article History:

Received: 24 January 2024
Revised: 15 March 2024
Accepted: 24 April 2024

Keywords— ***Nepali language, political profiling, vector space model, TF-IDF, word-embedding, word2vec, doc2vec, cosine similarity***

Abstract— **Everyday people in social networks create a huge amount of data as posts, blogs, tweets, articles, comments in form of text, images, audios and videos. The number of social media users and the data they are adding up in the cloud is increasing drastically day by day. People from all over the globe with different regions, cultures, languages, education, and public figures post or blogs reflecting their vision and opinion. These micro-blogs are now being used by researchers and business houses to analyze behaviour, sentiment and daily life-consuming habits, expenses capacity. In this paper, we are concerned about classifying a Nepali Twitter user to one of the pre-defined classes of political parties in Nepal using a vector space model. In this approach, a set of words is defined as a document class that represents a political party. A number of steps for text preprocessing are to be done based on the morphological structure of the Nepali language for a better result. TF-IDF and Doc2Vec methods are used to extract the feature of the terms used in tweets. Cosine similarity as a classifier is used to match the tweeter's profile with the political party's class and find the maximum similarity. Finally, compare the result between TF-IDF and Doc2Vec to conclude which one is more effective in the domain of tweets in the Nepali language.**

I. Introduction

According to Statistica.com [1] 2.62 billion people all over the globe are engaged in social media like Facebook and Twitter. The growth of people's engagement in society in this decade is amazing which is far much more than previously predicted. Every day people with different regions, culture, language, education, public figures posts or blogs regarding lifestyle, roaming, business, politics, weather, violence.

Social networks' users from different cultures and backgrounds post/tweet large numbers of textual arguments reflecting their vision/opinion/perspective 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’s opinion about the elections and result has attracted the interest of many researchers and the press. 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 was characterized by an intense competition [2]. The discussions of political conflict between Hillary and Trump were seen on the surface among the users of online social networks like Twitter.

Documents may be texts, music, or images where generally text is preferred. It involves in automating the classification of documents to different clusters regarding attitudes, opinions, emotions, ethnicities, religions. 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 used. The proposed model classify a twitter user to pre-defined class of political party according to the tweets he posts or retweets in his timeline.

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 Medias, personal conversations and messaging.

Due to complexity in data pre-processing in Nepali, less number of works has been done in text classification. There is no significant achievements achieved for political disclosure of tweeters using Nepali language.

The goal of this work is to present and compare the results obtained from the text classification for political profiling of Nepali twitter user using Cosine similarity with TF-IDF and Cosine similarity with Doc2Vec. Predicting a person's political affiliation using the tweets from his timeline is the main objective the study. For the comparison of TF-IDF and Doc2Vec we use the most popular text evaluation measures Accuracy, Recall, Precision and F-measure.

In doing the study a corpus of tweets dataset of different political parties in Nepali language is prepared. We scrap tweets from Nepali political leaders, ex-secretaries of Nepal Government, social activists and some renowned journalists which were published on Twitter up to December 11.

II. Literature Review

Text classification is done for the categorization of text (sentence/paragraph/articles) according to its words and its context. In Natural Language Processing (NLP), text classification is a primary tasks with broad applications such as Scalability, Real-time analysis, Consistent criteria, Sentiment Analysis, Summary of Text and Language Detection.

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 my enterprises to enhance decision-making and automate 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. In manual way, a person is responsible for understanding the context of text and categorizes it accordingly. The second one deploys machine learning, natural language processing to build up a model. It may be time consuming at first to train a machine but can automatically classify text in a faster and more cost-effective way. Combining these two a hybrid model can be derived for complex data.

A. Related Work

Josemar A. 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 are 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 are analyzed.

Bermingham and Smeaton [4] are also concerned with predicting electoral outcome, in particular, the outcome of the Irish General Election of 2011. Supervised classification with unigram features was used to analyze political sentiment in tweets achieving 65% accuracy on the task of positive/negative/neutral classification. It is concluded that sentiment plays an important role but volume is a stronger indicator of election outcome than sentiment.

Tarek Elghazaly et al. [5] used Support Vector Machine (SVM) and Naïve Bayesian (NB) to investigate the use of TF-IDF for developing document vector. They measure the accuracy and time to get the result for each classifier and determine which classifier is more accurate for Arabic text classification.

Shahi and Yadav [6] compares the classification techniques, the Naive Bayes and SVM, to evaluate which is better to classify Mobile SMS to ham or spam filtering for Nepali text.

Dangol and Timalsina [7] implement 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.

Kaushal 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.

B. Relevant Theory

1) Feature Selection and Extraction

Human brain is highly sensitive to pictures, graphs, sound but computer or machine takes numbers to compute. Natural Language processing 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 the features/ information from text data is vital technique which is basically used to reduce the dimension and identify the important features in a significant approach. The following techniques can be used for extracting features from text data.

2) 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 the 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 check if the word occurs or not in a group of words.

3) TF-IDF

One of the drawback of Bag of Words method is that the words with higher frequency becomes dominant in the document which may not provide much significant and efficiency for the model. Due to this problem domain specific words which does not have larger frequency may be ignored. Term Frequency – Inverse Document Frequency does what is expected, means it scales down the score of those words that are frequently common in all documents. It generate 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:

$$\mathrm{IDFw} = \log(\frac{\mathrm{N}}{\mathrm{DFw}}) \tag{1}$$

And finally Score or Weight of word TF-IDF is given by:

$$\mathrm{TF} - \mathrm{IDFw} = \mathrm{TFw} * \log(\frac{\mathrm{N}}{\mathrm{DFw}}) \tag{2}$$

Where,

TF is the number of occurrences of w in document d.

DF is the number of documents containing the word w.

N is the total number of documents in the corpus into consideration.

4) *Word2Vec*

Word2Vec [9] is a word embedding technique widely used for text extracting features for text classification. It is a two-layer neural networks results a semantic contexts 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 produces a vector space with hundreds of dimensions. Algorithmically, CBOW and skip gram both models are similar.

5) *Doc2Vec/Paragraph Vector*

Doc2vec also known as Paragraph Vector [10] 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 applicable to sentences, paragraphs, and documents of any length.

6) *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 use 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. It measures cosine of angle between the different dimensions of a word which are represented in an n-dimensional vectors. Mathematically, it is the dot product of the two vectors divided by the product of the magnitudes of the same vectors.

$$\text{Similarity}\ (\text{docA}, \text{docB}) = \frac{A.B}{|A| * |B|} \qquad (3)$$

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

III. METHODOLOGY

A. *Proposed Model*

The proposed model as shown in fig. 1 consists of the 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.

B. *Data Collection*

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 peoples' tweets with number of 254414 tweets of size 33.2 MB is scrapped from twitter. After the preprocessing of data 94195 tweets of size 19.3 MB were remained. Manual categorization was done to introduce 6 different categories according to political affiliation and corpus of tweets was built.

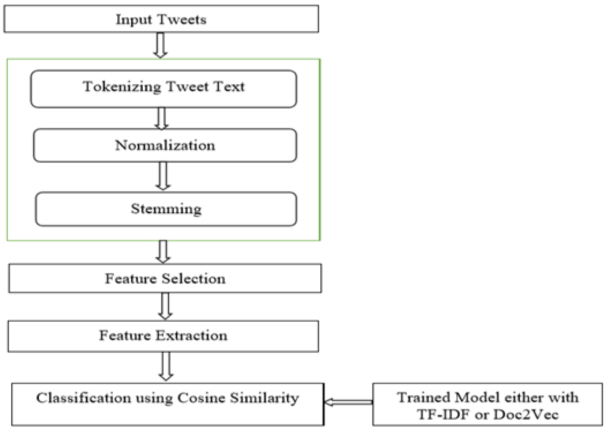


Fig. 1. Proposed model of the system

Tweets of top level political leader are taken as the training and testing data for respective class of political parties. Some renown personalities such as journalist, activist, authors are considered as 'non-political' category during the feature extraction phase in order to improve accuracy of the model.

TABLE I. TWEETS DATA DISTRIBUTION

Category	No of Tweeters	No of Tweets	Average no of Tweet By Individual	No. of Tweeters Tweeting more than average tweets
Nepali Congress	13	11,992	922	7
Nekapa	25	24,437	977	12
Madhesbadi	6	8,642	1,440	4
New Parties	14	24,640	1,760	9
Raprapa	5	9,192	1,838	4
Non-Political	9	15,292	1,699	5
Total	72	94,195	1,255	41

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 tab 1. As TF-IDF and Word2Vec both are supervised technique of Machine learning, they need pre-defined political classes to for the testing of tweeters' profile. So tweets of First level political leaders' tweets have been taken. Our biasness may occur only in the tweets that are taken as non-political category. So Manual categorization is done to classify the training data in which 6 classes are defined as Congress, Nekapa, Madhesbadi, New Parties, Raprapa and Non-Political categories.

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" [11]. All the characters expects Punctuations, digits are taken into consideration. As the study is focused to Nepali language tweets, Nepali language based morphological tasks are done. First one is to unify the rhaso-dirgha ekar and ukar, स and श, ब and व. Removing the tense, adjective, plural, gender related suffixes from the words. Finally stop words are 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 ie party tweets can be generated.

The machine learning algorithms are used for classification of Individual Profile's Tweet to one of the seven category defined. The main aim of ML algorithms/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. The algorithm assign the person to that category to which the cosine similarity is greater.

1) Algorithm

The proposed model has following steps:

Step 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!उपउप-चुनावले नेकपालाई स्पष्ट सन्देश दिएको छ-सरकारको शैली सच्याउ नत्र सकिन्छौ सच्चिने वा सक्किने उसको कुरा!-उप-चुनावले नेकपालाई स्पष्ट सन्देश दिएको छ-सरकारको शैली सच्याउ नत्र सकिन्छौ सच्चिने वा सक्किने उसको कुरा!चुनावले नेकपालाई स्पष्ट सन्देश दिएको छउप-चुनावले नेकपालाई स्पष्ट सन्देश दिएको छ-सरकारको शैली सच्याउ नत्र सकिन्छौ सच्चिने वा सक्किने उसको कुरा!-उप-चुनावले नेकपालाई स्पष्ट सन्देश दिएको छ-सरकारको शैली सच्याउ नत्र सकिन्छौ सच्चिने वा सक्किने उसको कुरा!सरकारको शैली सच्याउ नत्र सकिन्छौ सच्चिने वा सक्किने उसको कुराउप-चुनावले नेकपालाई स्पष्ट सन्देश दिएको छ-सरकारको शैली सच्याउ नत्र सकिन्छौ सच्चिने वा सक्किने उसको कुरा!!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", "कुराउपउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुराउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुराचुनावलेउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरानेकपालाईउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरास्पष्टउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरासन्देशउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरादिएकोउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुराछउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरासरकारकोउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुराशैलीउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरासच्याउउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरानत्रउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरासकिन्छौउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरासच्चिनेउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरावाउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरासक्किनेउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुराउसकोउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा", "उप-चुनावले", "नेकपालाई", "स्पष्ट",

"सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुराकुराउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुराउप-चुनावले", "नेकपालाई", "स्पष्ट", "सन्देश", "दिएको", "छ", "सरकारको", "शैली", "सच्याउ", "नत्र", "सकिन्छौ", "सच्चिने", "वा", "सक्किने", "उसको", "कुरा","!".

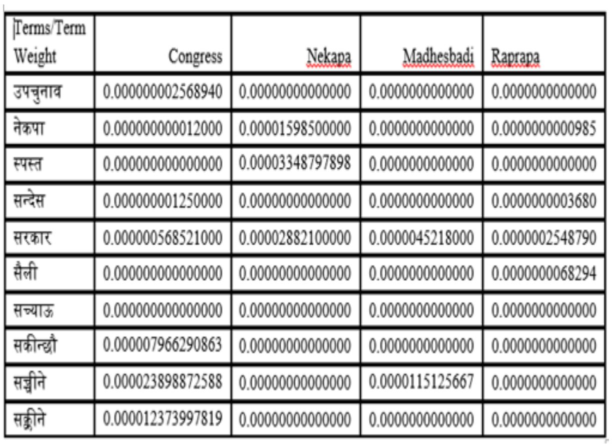
Step 2 : Replace Bartshya (ट, ठ, ड, ढ ण) to Dantya (त, थ द, ध, न), श to स, all Rhasyawo Ekar, Ukar to Dirgha. Remove Purnabiram, Halanta. The tokens changed to "उपचुनावले", "नेकपालाई", "स्पस्त", "सन्देस", "दीएको", "छ", "सरकारको", "सैली", "सच्याऊ", "नत्र", "सकीन्छौ", "सच्चीने", "वा", "सक्कीने", "उसको", "कूरा","!".

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".

Step 3 : 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".

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 extraction process. TF-IDF generated a single feature for each term whereas Word2Vec generates several hundred normally (100-300) dimensional features for each term. 300 dimension is used in proposed model. The TF-IDF of the Gagan Thapa's tweet is shown in table II.

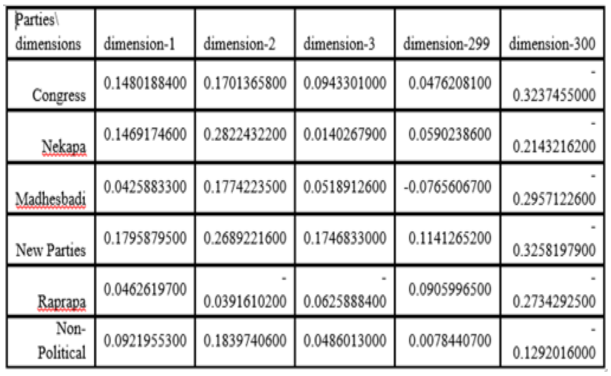
TABLE II. WEIGHTS OF TERMS FOR DIFFERENT POLITICAL CLASSES EXTRACTED BY TF-IDF MODEL



Terms/Term Weight	Congress	Nekapa	Madhesbadi	Raprapa
उपचुनाव	0.000000002568940	0.0000000000000	0.000000000000	0.000000000000
नेकपा	0.000000000012000	0.00001598500000	0.000000000000	0.000000000985
स्पस्त	0.000000000000000	0.00003348797898	0.000000000000	0.000000000000
सन्देस	0.000000001250000	0.0000000000000	0.000000000000	0.000000003680
सरकार	0.000000568521000	0.00002882100000	0.0000045218000	0.0000002548790
सैली	0.000000000000000	0.0000000000000	0.000000000000	0.000000068294
सच्याऊ	0.000000000000000	0.0000000000000	0.000000000000	0.000000000000
सकीन्छौ	0.000007966290863	0.0000000000000	0.000000000000	0.000000000000
सच्चीने	0.000023898872588	0.0000000000000	0.000115125667	0.000000000000
सक्कीने	0.000012373997819	0.0000000000000	0.000000000000	0.000000000000

Step 4 : 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 as tab. 3. 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 III. SET OF FEATURES OF POLITICAL CLASSES GENERATED BY AVERAGING THE WEIGHT FROM WORD2VEC



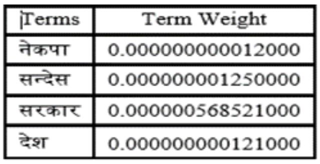
Parties\ dimensions	dimension-1	dimension-2	dimension-3	dimension-299	dimension-300
Congress	0.1480188400	0.1701365800	0.0943301000	0.0476208100	- 0.3237455000
Nekapa	0.1469174600	0.2822432200	0.0140267900	0.0590238600	- 0.2143216200
Madhesbadi	0.0425883300	0.1774223500	0.0518912600	-0.0765606700	- 0.2957122600
New Parties	0.1795879500	0.2689221600	0.1746833000	0.1141265200	- 0.3258197900
Raprapa	0.0462619700	- 0.0391610200	- 0.0625888400	0.0905996500	- 0.2734292500
Non-Political	0.0921955300	0.1839740600	0.0486013000	0.0078440700	- 0.1292016000

Step 5 : Compute the document vector of test documents following the above all steps.

Consider a timeline of a tweeter containing the words "नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"नेकपा"नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"", ""नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"सन्देस"नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"", ""नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"सरकार"नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"", ""नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"सैली"नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"", ""नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"सच्याऊ"नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"", ""नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"सकीन्छौ"नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"", ""नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"सच्चीने"नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"", ""नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"सक्कीने"नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"""नेकपा", "सन्देस", "सरकार", "सैली", "सच्याऊ", "सकीन्छौ", "सच्चीने", "सक्कीने"

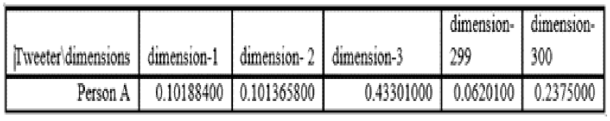
After preprocessing, data are shown as in Table IV.

TABLE IV. WEIGHTS OF TERMS CONTAINED IN TEST DOCUMENT EXTRACTED BY TF-IDF MODEL



Terms	Term Weight
नेकपा	0.000000000012000
सन्देस	0.000000001250000
सरकार	0.000000568521000
देश	0.000000000121000

TABLE V. SET OF FEATURES OF TERMS CONTAINED IN TEST DOCUMENT EXTRACTED BY DOC2VEC



Tweeter\dimensions	dimension-1	dimension- 2	dimension-3	dimension-299	dimension-300
Person A	0.10188400	0.101365800	0.43301000	0.0620100	0.2375000

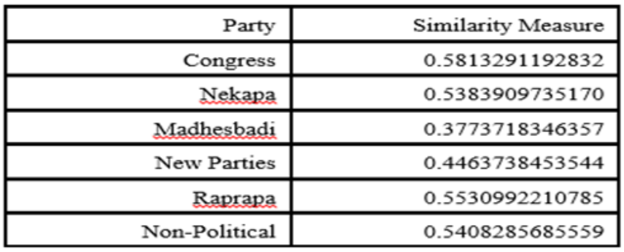
Step 6 : Evaluate the similarity measures between the test document and the document class i.e. political party's document using cosine similarity as Tab. 5.

Step 7 : Classify the test document to the political party that gives the maximum similarity measure. For a text of tweets from a Congress leader's timeline, cosine similarity gives the result as shown in Tab.6.

Step 8 : 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 whereas 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. Considering 10 fold in phase 1, two of the folds contains 8 tweeters and remaining 8 folds contains 7 tweeters.

The tweet data of seven folds are used as the training data and the one fold's data is taken as test data. The tweeters in the test fold is tested one by one with model and find out maximum similarity measure with all political class as classified by cosine similarity and assigned to the class having maximum similarity measures

TABLE VI. RESULT OF CLASSIFICATION OF THE TWEETER'S PROFILE BY COSINE SIMILARITY USING THE TF-IDF MODEL



Party	Similarity Measure
Congress	0.5813291192832
Nekapa	0.5383909735170
Madhesbadi	0.3773718346357
New Parties	0.4463738453544
Raprapa	0.5530992210785
Non-Political	0.5408285685559

IV. RESULT AND EVALUATION

Performance metrics are based on the classification done by Cosine similarity using the features developed by TF-IDF or Doc2Vec models in which 10-fold and 5-fold cross validation method is used for testing. Evaluation measures used are Accuracy, Precision, Recall and F1-Score.

A. *Four outcomes of classification*

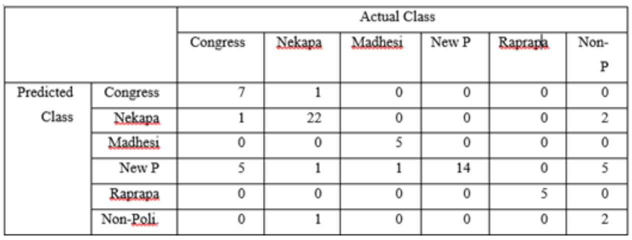
A classifier predicts all tweets data instances of a test dataset as either positive or negative. Any of the four outcomes true positive, true negative, false positive and false negative will be predicted. True positive and true negatives are the correctly classified observations.

Based on the produced outcomes, the following Performance Metrics can be calculated.

B. *Confusion Matrix*

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 is done validation is performed. The result of all the experiments give the following results. Confusion Matrix was created with 10-fold validation method as shown in Tab. 7. Accuracy, Precision, Recall and F1 Score are used for the evaluation of the classifier. Precision of a class measures the ratio of correctly classified document to the total number of documents classified in the class. Recall is measure of ratio of correctly classified document to total document in a class. F1 score is calculate as the harmonic mean formula using precision and recall.

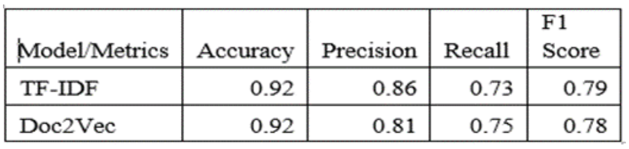
TABLE VII. MULTI-CLASS CONFUSION MATRIX OF WITH 10-FOLD CROSS VALIDATION



		Actual Class					
		Congress	Nekapa	Madhesi	New P	Raprapa	Non-P
Predicted Class	Congress	7	1	0	0	0	0
	Nekapa	1	22	0	0	0	2
	Madhesi	0	0	5	0	0	0
	New P	5	1	1	14	0	5
	Raprapa	0	0	0	0	5	0
	Non-Poli	0	1	0	0	0	2

Then, the calculations of performance measures are straightforward once the confusion matrix is created. Here, Cosine similarity is used for measuring the similarity between the political party category and individual's test tweets for classification. The TF-IDF and Word2vec model is trained first using 90 percent of profile's tweets and after that 10 percent of Tweets are used as testing purpose. It is done using K-Fold cross validation method. Result of all the experiments are listed in the Tab. 8.

TABLE VIII. COMPARISON OF PERFORMANCE METRICS USING TF-IDF AND DOC2VEC MODELS



Model/Metrics	Accuracy	Precision	Recall	F1 Score
TF-IDF	0.92	0.86	0.73	0.79
Doc2Vec	0.92	0.81	0.75	0.78

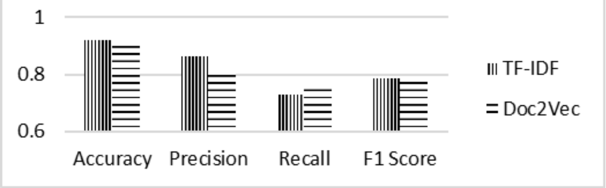


Fig. 2. Comparison of Performance metrics using TF-IDF and Doc2Vec Models

V. DISCUSSION AND ANALYSIS

In this research work, tweets from twitter are used for training and testing data to perform a political profiling. A corpora for 6 political parties is 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. The metrics of the comparison of the TF-IDF and Doc2Vec are the most popular text evaluation measures Accuracy, Precision, Recall and F-measure.

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. Results with TF-IDF is slightly promising in comparison to Doc2Vec. Due to small size of dataset, neural network algorithms like Doc2Vec was not able to perform better as compared to traditional machine learning algorithms like TF-IDF. 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 as shown in fig. 2. A Corpora of Nepali tweets is developed which can be made available for use by other researchers.

VI. CONCLUSION AND FUTURE ENHANCEMENT

The result shows that TF-IDF beats the Doc2Vec method. Theoretically, Doc2Vec performs better than TF-IDF due to the reason that Doc2Vec also considers the context of terms while TF-IDF only considers frequency of terms in content. So more tweets can be collected from twitter in future and perform the analysis. State of the Art algorithms such as deep learning algorithms can be deployed for achievement of better result. A large dataset is to be built because deep learning algorithms need huge training data for better learning of model. 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 Nepal language which leads to better performance of classification done in this study.

REFERENCES

[1] Number of social media users worldwide from 2010 to 2021 (in billions) https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/ Accessed 20 June 2019.

[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 20 June 2019.

[3] Josemar A. Caetano, Hélder S. Lima, Mateus F. Santos and Humberto T. MarquesNeto "Using sentiment analysis to define twitter political users' classes and their homophily during the 2016 American presidential election", Journal of Internet Services and Applications (2018)

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

[5] Tarek Elghazaly, Amal Mahamoud, Hesham A. Hefnu. "Political Sentiment Analysis Using Twitter Data." ICC '16, Cambridge, United Kingdom. (2016)

[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. (2013)

[7] Dinesh Dangol and Arun K. Timalsina. "Effect of nepali language features on nepali news classification using vector space model." (2013)

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

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

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

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

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

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