Mining multilingual and multiscript Twitter data: unleashing the language and script barrier

Bidhan Sarkar

Department of Computer Science and Engineering,
National Institute of Technology,
Durgapur, West Bengal, India
Email: sarkarbidhan59@gmail.com

Nilanjan Sinhababu

Department of Computer Science and Engineering,
Sanaka Educational Trust's Group of Institutions,
Durgapur, West Bengal, India
Email: nilanjan.thecseboy@gmail.com

Manob Roy, Pijush Kanti Dutta Pramanik and Prasenjit Choudhury*

Department of Computer Science and Engineering,
National Institute of Technology,
Durgapur, West Bengal, India
Email: manobroy92@gmail.com
Email: pijushjld@yahoo.co.in
Email: pcnitdgp@gmail.com
*Corresponding author

Abstract: Micro-blogging sites like Twitter have become an opinion hub where views on diverse topics are expressed. Interpreting, comprehending and analysing this emotion-rich information can unearth many valuable insights. The job is trivial if the tweets are in English. But lately, increase in native languages for communication has imposed a great challenge in social media mining. Things become more complicated when people use Roman scripts to write non-English languages. India, being a country with a diverse collection of scripts and languages, encounters the problem severely. We have developed a system that automatically identifies and classifies native tweets, irrespective of the script used. Converting all tweets to English, we get rid of the 'script vs language' problem. The new approach we formulated consists of Script Identification, Language analysis, and Clustered mining. Considering English and the top two Indian languages, we found that the proposed framework gives better precision than the prevailing approaches.

Keywords: Twitter mining; language classification; script identification; Indic language; preprocessing; naive Bayes; support vector machine; LDA.

Reference to this paper should be made as follows: Sarkar, B., Sinhababu, N., Roy, M., Pramanik, P.K.D. and Choudhury, P. (2020) 'Mining multilingual and multiscript Twitter data: unleashing the language and script barrier', *Int. J. Business Intelligence and Data Mining*, Vol. 16, No. 1, pp.107–127.

Biographical notes: Bidhan Sarkar received his MTech degree in High Performance Computing (Computer Science and Engineering) (2017) from NIT Durgapur. He is passionate about the new emerging technologies and his research interests include data analytics, machine learning, and artificial intelligence. He is currently working in research areas such as social media data analytics, data mining, etc.

Nilanjan Sinhababu is a BTech Graduate who is very much passionate about the new evolving technologies and like to be creative. As a proficient programmer and expert in most basic languages as well cutting edge technologies such as machine learning and data analytics. He is actively engaged in research in the areas of recommendation systems, image analysis, big data analytics, etc.

Manob Roy is a PhD Research scholar in the Department of Computer Science and Engineering at National Institute of Technology, Durgapur, India. He is a Gold Medallist in Computer Science during his graduation and has completed MCA. He is highly passionate about the emerging field of Big Data Analytics and has deep interests in its technologies. He is actively involved in the research areas of social media analytics, recommendation systems, data mining, etc.

Pijush Kanti Dutta Pramanik is a PhD Research scholar in the Department of Computer Science and Engineering at National Institute of Technology, Durgapur, India. He has acquired a range of professional qualifications in the field of information technology namely MIT, MCA, MBA (IT), MTech (CSE), and MPhil (CS). He is actively engaged in research in the domains of internet of things, grid computing, fog computing, crowd computing and recommendation systems. He is passionate about technical writing.

Prasenjit Choudhury is an Assistant Professor in the Department of Computer Science and Engineering at National Institute of Technology, Durgapur, India. He has completed his PhD in Computer Science and Engineering from the same institute. He has published more than 40 research papers in international journals and conferences. His research interest includes wireless network, data analytics, and recommendation systems.

1 Introduction

The online social network has brought a revolution in the way human communicate. It has become an inseparable part of life, most often an obsession, for the majority of internet users. Since 2004, the number of social media users has grown near exponentially and has stretched to 2.8 billion globally (http://www.adweek.com/digital/the-growth-of-social-media-from-trend-to-obsession-infographic/; http://www.marketwired.com/press-release/new-research-reveals-global-social-media-use-increased-by-21-percent-in-2016-2190591.htm). In India, the number has reached to formidable 115 million (http://timesofindia.indiatimes.com/business/india-business/90-of-new-net-

users-non-english/articleshow/58371769.cms). People are expressing themselves on social media like never before in the pre-'social media age'. One reason may be that they find it lot easier to vent their feeling as it is less uncomfortable compared to the face-to-face interaction. Even the things one never could think of articulating in person is doing that without any timidity through the online platforms. Social networks have been very much instrumental in spreading our thoughts, perceptions, and concerns over different topics including products, services, events and people to a sizable audience. In particular, the micro-blogging sites, e.g., Twitter have become the favourite platform for major online discussions and gradually turned into a massive opinion storing hub. On an average, around 6,000 tweets are tweeted on Twitter every second which accounts to 500 million tweets per day and the number is continuously growing at around 30% per year (http://www.internetlivestats.com/twitter-statistics/). In India, it is supposed to grow at an average rate of 14% over the next three years, (https://www.statista.com/statistics/303691/twitters-annual-growth-rate-in-india/) and totalling to roughly 30 million during that period (https://www.statista.com/statistics/381832/twitter-users-india/). The mass opinions can play a significant role in shaping events. Proper analysis of this emotion-rich information can be utilised for many real-world purposes. Rule makers can sense the socio-political development and movement and deal accordingly. Businesses can observe products/services consumption trends and find new patterns that can influence their marketing and business planning and decisions. The accumulated social data can be used to predict the flu or a disease outbreak. It may also be very effective in crises management.

To seize the benefits, the first job is to read and interpret the messages. It is straightforward if the messages are written in English. But the popularity of usage of English in social media is witnessing a steady downfall since the past couple of years because of the influence of the native languages (Bhargava et al., 2016). In the context of India, despite a very rich diversity of languages (Bhargava et al., 2016; Sharma et al., 2015), the online language has always been dominated by the English known users. But thanks to cheap smartphones and affordable data prices, the number of internet users in India is growing at a very rapid speed. As a result of this, the Indian languages such as Hindi, Bengali, Tamil, Marathi, Telugu, Kannada, Gujarati and Malayalam are set to dominate the internet space. According to a recent study (http://timesofindia.indiatimes.com/business/india-business/90-of-new-net-users-non-english/articleshow/58371769.cms) by Google and KPMG, by 2021, Hindi will surpass English as the most online used language in India whereas Bengali, Marathi, Tamil, and Telugu will form 30% of the total local language user base in the country. At present, the study suggests, as compared to 175 million English users, 234 million internet users in India uses native languages and the number is supposed to grow to 534 million in the next four years. In fact, 90% of the new internet users in the country do not prefer English.

So, it is very crucial to have interpreters that can interpret the local languages used in social media. The language identification has already been a great challenge for quite a long time (Gold, 1967). The situation has been aggravated as most native languages speaking people prefer to use English alphabets (Latin/Roman script) to write in social media because, perhaps, that is easier to type (Bhargava et al., 2016). And more often than not they use multilingual words in a single sentence. As a result, it has become extremely difficult to elucidate the messages. Hence, in spite of huge potential, the attempts of mining information from social media data has been limited (Lin and Ryaboy,

2013) due to the fact that neither proper script analysis mechanisms nor standard transliterating methods are available (Cardoso et al., 2016) to handle this multilingualism. In recent years, however, it has become an active area of research as many machine learning approaches have accelerated the success rate in the opinion mining of micro-blogs.

In this paper, we have proposed a novel method to mine Twitter opinions even if they are written in multiple languages and scripts. We have experimented with the Indic languages Bengali and Hindi, written either in their own script or in Roman scripts or a combination of both along with usual English. We have concentrated on Tweeter data because tweets are brief and therefore they tended to be specific. Hence it is easy to extract the connotation.

Our approach is to, first identify the script of the tweet, then analyse the language of it and then convert all the tweets into English irrespective of the language and script they are written in. If it is found to be Indic script, the language is identified and the texts are directly translated into English. If the script is Roman then it is determined whether the language is English or Indic. If it is Indic, the texts are transliterated and translated into English subsequently. So, ultimately every text irrespective of the language and script are transformed into English with the Roman script, which is then mined using traditional mining techniques. In our approach, we have considered the whole sentence as a single feature and performed a sentence level classification which implies, a particular single sentence conforms to a particular language and script. We have devised an algorithm for language processing which derived from the machine learning algorithms viz. naive Bayes classifier and SVM.

The paper is organised as follows: The exact problem that we are trying to deal with is defined in Section 2. In Section 3, we reviewed related research in the area of language classification. Section 4 explains the background methods related to our approach along with the dataset. In Section 5, we explained the methodology with different phases that have been performed for the task, along with the respective results obtained are described in details. Section 6 concludes the paper.

2 Problem definition

Language recognition in Twitter takes place through the metadata support that the Twitter possesses inherently. When any analysis in Twitter mining is to be done, the insights provided by the metadata plays a vital role. Unfortunately, when Indian Twitter is concerned, the accuracy of the language identification is not much when compared to other European and US languages as can be seen from Table 1.

Table 1 Performance comparison

Roman_Indic Tweets	*Twitter_lang*	*Actual_language*
ami valo achi tumi kemon acho	‘es’ (Spanish)	Bengali
kichu korar nai ar thamiye lav ki	‘hi’ (Hindi)	Bengali
amir bhai kabhi aoo na Khyber pakhtoonkhwa hamara mehmaan bannn kay	‘in’ (Indonesian)	Hindi
plz sir ak moka dedo Bhagaw ban ke apko nirash nhi karuga pese nhi chahiye bs ak moka hath jod ta Bhagaw sun lo ak Ripley abhishek	‘tl’ (telegu)	Hindi

The metadata is basically achieved in JSON format and the meta-data contains various fields along with the 'text' field, 'location', location of the user, 'Twitter_lang', this is the language field provided by the Twitter and finally 'id', which is the system generated object id of the user, 'Detected_Script', which has been additionally added for the research for detecting scripts using Unicode. A couple of snapshots regarding the metadata-based approaches are envisaged in Tables 2 and 3.

2.1 Successful detection

While inspecting the tweets from the JSON file we know as discussed above that it provides many metadata fields that give us immense insights about the tweets. We selected the necessary fields required for our research and performed a scrutiny. Many of the classified tweets provided correct identification according to the intended language. Following is an instance where 'Twitter_lang' successfully identifies an English text as 'en' (left) and a Hindi text as 'hi' (right). The sample is shown in Table 2.

Table 2 Successful cases

'_id': ObjectId '58a3f 27ee2aa940d481f', 'location' : 'Thiruvalla, Kerala, India', 'Detected_Script' :'English', 'Twitter_lang': 'en', 'Text': 'ISRO sends record 104 satellites in one go breaks Russias record'	'_id' : ObjectId ''58a3f 2dde2aa940d481f ', 'location': 'New Delhi', 'Detected_Script': 'English', 'Twitter_lang': 'hi', 'Text': 'ye kaisi bhasha likhte hain aap isko delete karen varna delete karna hoga thoda viewership survey padhen aankh khulegi'

2.2 Unsuccessful detection

But the above situation seldom happens and mostly the cases that actually happen is that the Indic languages are misinterpreted as some European or American languages and henceforth all the classification and interpretation of the tweet drastically changes. In the following instance, instead of detecting as 'hi', the 'Twitter_lang' are detected as 'tl' (left) and 'in' (right). The details are shown in Table 3.

Table 3 Unsuccessful cases

'_id': ObjectId '58ac21e8e2aa94102ca274c0', 'location': 'Indore', 'Detected_Script': 'English', 'Twitter_lang': 'tl', 'Text': 'plz sir ak moka dedo Bhagaw ban ke apko nirash nhi karuga pese nhi chahiye bs ak moka hath jod ta Bhagaw sun lo ak Ripley abhishek'	'_id': ObjectId '58ac23e1e2aa94102ca276cb', 'location': 'Ranchi, India', 'Detected_Script': 'English', 'Twitter_lang': 'in', 'Text': 'Bc jaa tu usi dam mein khud ke marr jaa'

3 Related work

This section is focused on all the ongoing efforts in the domain of Twitter mining, language identification, topic clustering. Researchers across the globe have found the trend to use native language for online communication is gaining immense preferences. There has been little work on automatic languages in social media and rarely any work in Indian context. A significant effort in the linguistic literature to find the statistical data of

the social media has been done by Hong et al. (2011), in their research to find the top 10 most popular languages on Twitter. The research work evaluated that out of 62 million tweets, only half of the tweets were in English, thus justifying the scope for the research in native languages. While language classification was on the peak, Gottron and Lipka (2010) and Laboreiro et al. (2013) cited that naïve Bayes classifier provided huge success in language classification in European languages. Barman et al. (2014) and Bhargava et al. (2016) have used nonlinear classifier (SVM) in the research to find that it provides magnificent results but mostly focused on the sentiment analysis rather than on language classification. Word level classification has been the trend for quite a time (Dutta et al., 2015; Banerjee et al., 2015) but most of the work does not concern the working of Twitter as the tweets are generally short texts in a single language, hence depriving the need of code mixing. Truica et al. (2015) had proposed a statistical method, that detect and classify Twitter data and news articles based on dictionary of stop words and diacritics automatically. The limitation of the above technique is languages might have common words, at that time the detection will be not correct, but such problems are mitigated to a large extend in our technique. King and Abney (2013) took the dataset using a data crawler freely available Boot Cat, which crawls data from various site based on the searched keywords and the tuples that are being created based on the searched keywords. N-gram is a popular natural language processing which is being used independently and along with various classification approaches, as implemented in Singh and Goyal (2014). Gottron and Lipka (2010) used different machine learning approaches such as naïve Bayes in combination with n-gram for the feature extraction while training the model. There been some work done by Lui and Baldwin (2011, 2014) regarding the language identification related to Twitter data and as well as data from various web pages using tool called Langid but not considering the case of Indian languages. Langid basically uses naïve Bayes as the classifier which the mostly used and has been a well-known classifier since the past.

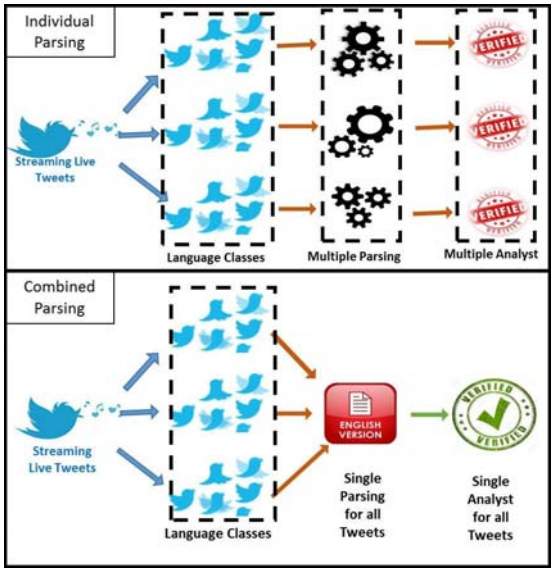
India being a multi-script nation, language classification has considerable attention and focus for Indian languages. Barman et al. (2014) compared different language identification approaches including dictionary-based, supervised word level classification using SVM and in addition, sequence labelling using conditional random field (CRF) model and finally concluded that CRF model outperforms the other approaches, CRF model is being used by few other researchers also Chittaranjan et al. (2014), Banerjee at al. (2015), Dutta et al. (2015), King and Abney (2013) but as per the survey CRF model is not suitable in our case as it is very difficult to re-train the model every time and moreover it does not work with unknown words which is not present in the training data set, so it means it requires large enough training data to get the best result as per the requirement. Bhargava et al. (2016) developed a system for sentiments mining of code mixed sentences for English with combination of four other Indian languages (Tamil, Telugu, Hindi and Bengali), in two phases. These stages includes language identification and sentimental analysis using different machine learning tools, with Google transliteration in between to convert all the texts to their respective regional languages. But the need of separate parsers and language experts for each regional language makes it a very divergent approach. They did classification at the word level, setting few sets of words belonging to respective languages as training sample data. They used n-grams after using naive Bayes classifier for getting more accuracy at the word level. There have been a few works done in topic clustering domain such as Hong and Davison (2010), Mehrotra et al. (2013) and Zou and Song (2016) mainly by unsupervised approaches using LDA as

their basic tool and further naming the topic by themselves that is mostly related to the cluster obtained.

4 Research background

The strategy that is implemented in this research work is that it follows a converging approach in the entire classification system which in term makes it very easy vis-à-vis separate parsing of individual languages. The obvious complexities that would arise due to separate parsing are that every language will require its own parser for later analysis purposes and also individual language analysts will be required for cross-validation of the separate clusters. Whereas if all the tweets are translated into a single language at the very beginning, then all the later efforts for analysis gets simplified and less resource consuming. Figure 1 brings out the subtle differences that are important to understand the system before delving in any further into it.

Figure 1 Comparison of classification approach (see online version for colours)



The objectives of this paper can be divided into three major parts viz. language classification of Indic tweets, channelling all the tweets into English irrespective of the script or intended languages for further processing and finally clustering all the tweets into relevant topics for further mining. Machine learning techniques are used to train models in the language classification phase (Lui and Baldwin, 2011) and later on the combination of open-source transliteration and translation are used to convert all the tweets into English. The architectural framework of the proposed system is shown in

Figure 1. The research approach includes a comparison of the top machine learning tools and extracting beneficial features from both. Following are the individual explanations of components used in the architecture. Every component is described with its pros and cons, as and when necessary.

4.1 Corpus acquisition

The corpus for our research is generated live from Twitter through python package 'tweepy' for streaming live tweets. Our target was focused on the acquisition of data for the top two highly spoken languages in India, viz. Hindi and Bengali. We collected 3,000 live tweets each in Devanagari and Bengali scripts. Later through experts' intervention in case of Roman script tweets, we further classified 3,000 tweets each in Hindi, Bengali and English languages to cover the all possible 'script vs language' complexities. Hence, we had a rich data set which covered absolute English tweets, all those who used native script for expressing their views in Twitter and also those who used Roman scripts to express their views in their local languages. Our corpus was then filtered for the punctuation marks in the pre-processing stage, but we kept the emo-signs intact for the emotional analysis of the acquired data through topic clustering.

Henceforth, the data was crosschecked by human experts in order to make sure that proper data is allotted to the pre-defined partitions created during the data acquisition phase. Once the data is properly checked, individual algorithms are run on the datasets previously prepared and post analysis consequences are once again evaluated by human experts.

4.2 Tools and resources

The mode of research pursued by us consisted of thorough python programming, compiling Open Source APIs and multi-stage training of the data set. For the live streaming of the Twitter data, the Twitter streaming API 'tweepy' in used to fetch the data and is stored in MongoDB database connected through 'pymongo' with the python code. While streaming the data, UTF-16 character encoding identifies the scripts of the tweets and classifies them accordingly. The partitioned data sets are trained through node.js platform using the Limdu machine learning framework. Once training is complete, the trained data is evaluated by language experts through an interface built in Java Netbeans.

4.3 Script and language detection

The Unicode detection analysis is used for classifying the tweets to their respective scripts. We considered top three mostly communicated language in Indian Twitter, English, Hindi and Bengali for the purpose of analysis. Each script has its own Unicode range, so comparing the Unicode values of the tweets with those in any known range, scripts detection is performed which makes the filtering of the tweets easier for the system performance. For the research purpose, two-letter words are used to symbolise the scripts as follows:

- 'Rm' (to denote Roman) for English scripts
- 'Dv' (to denote Devanagari) for Hindi scripts
- 'Bn' for Bengali scripts.

But our paper also consists of research on intended languages and for that we will use 3-letter words to symbolise them as follows:

- 'Eng' for English language
- 'Hin' for Hindi language
- 'Ben' for Bengali language.

4.4 Machine learning classification techniques

Machine learning classification technique is generally of two types: linear and nonlinear classification. The language classification of Twitter data will be English, Hindi or Bengali, but it cannot give more than one solution at a time. So, the classification can be linearly separated into different classes. For this reason, the linear classifiers are more preferred over nonlinear classifiers to solve the above problem. Another important aspect of learning is its nature of learning, i.e., supervised or unsupervised (Ertel, 2011). We have adopted supervised learning, i.e., we have predefined the classification criteria. The architecture of our learning model is explained below with a flow diagram.

4.4.1 Naive Bayes classifier

Naive Bayes classifier is one of the oldest and among the most successful known algorithms for learning to classify text documents. The Bayesian classification is as a probabilistic learning method (naive Bayes text classification). It assumes every feature is independent of each other. To assign labels for every input vector, features are utilised using the formula below (Sharma et al., 2013):

$$P(label \mid features) = \frac{P(label) * P(features \mid label)}{P(features)} \tag{1}$$

Label in the above equation gives the classified language, i.e., Hindi, Bengali or English, and features are the words which have been extracted from the tweets.

4.4.2 Support vector machine

It is a learning system which utilises hypothesis space in high dimensional feature space. It is more considered where the number of samples is smaller than the number of features. SVM is a type of supervised learning technique which, in language detection, involves training a detection classifier via the frequency of various words appearing in a tweet.

Basically, SVM does not work with textual data; it is generally applied to numerical data. So, for converting the texts into numerical data, we have used n-gram of letters, before using it all the texts are being converted into lower case. In our case n = 3, each consecutive three letters are converted into either 0 or 1 based on its presence in the sentence, if it is present then it will be 1 or otherwise 0. The details are described in Table 4.

Table 4 SVM working principle

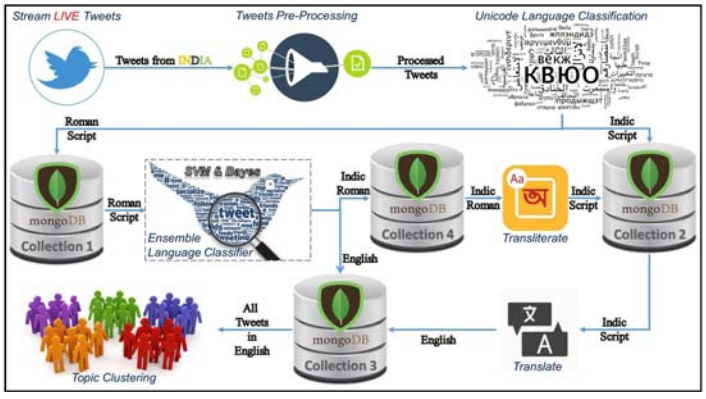
Tweets	*3-gram of letters*
'ami valo achi'	'ami', 'mi_', 'i_v', 'val', 'alo', 'lo_', '_ac', 'ach', 'chi'
'main accha hu'	'mai', 'ain', 'in_', 'n_a', '_ac', 'acc', 'cch', 'cha', 'ha_', 'a_h', '_hu'

5 Solution methodology

The steps involved in the whole process of script free mining:

- The first step involves streaming of live tweets from India using Twitter Streaming API.
- The tweets are then pre-processed on the fly by removing unnecessary characters in the pre-processing phase.
- Using unicode language identification, tweets are separated on the basis of their script and each script is stored in distinct MongoDB collection as
  - a collection 1: Roman script (which contains English as well as other languages that are written in Roman script like Bengali and Hindi, we named them as 'Indic Roman')
  - b collection 2: Indic scripts (Bengali script + Hindi script).
- Using ensemble language classifier (SVM + Bayes) all the tweets from collection 1 classified as 'English' are stored in Collection 3 whereas classified 'Indic Roman' are stored in collection 4 with a language tag.
- Our self-made API in collaboration with Google Transliterate is used to transliterate tweets in collection 4 which have language tag as Hindi or Bengali to convert them to their original script(Indic Script) and then they are stored in collection 2 after transliteration phase is complete.
- By using Google Translate API all the contents of collection 2 are translated to English and are stored in collection 3.
- Now since all the tweets are in English we can analyse the data on collection 3 for topic clustering using LDA ensembled with our personalised clustering system.

Figure 2 Proposed system architecture (see online version for colours)



We have used database collections in between phases to avoid bottleneck of data in between the phases. Since all the phases have different lifecycle we needed a buffer to make the steaming flow of the whole process.

Instead of only evaluating live data the tweets are stored in buffers (database collections in between phases) so as to examine time-line analysis of tweets.

5.1 Evaluation of tweets

We have performed analysis by streaming live tweets using python APIs that are being extracted from Twitter by bounding the location within India in order to get tweets intended in native languages. Twitter metadata provides many fields, but only the fields which are of significant relevance are considered in the work. So, the focus mainly lies in the metadata field 'Twitter_lang' which is key elements of all the future system performances because when the detection itself fails, the detection result provided by the Twitter cannot be taken under consideration for further mining. The collected tweets are raw in nature and are not fit for healthy analysis of the data. Hence, they are forwarded to the preprocessing phase.

5.2 Data preprocessing

Text preprocessing is a subtle phase for the later phase of text mining for data analysis. Data for analysis needs to be noise free as they are responsible for the deterioration. Preprocessing of the tweets is essential not only for the purpose of language analysis, but also for the ease of cross-validation of the system performance by the human experts operating at different levels of the process. Preprocessing in this paper includes the following processing:

- removal of website URLs as they are not relevant in our paper
- removal of hash tags (#) to have clear tweets for human experts engaged in manual verification
- removal of re-tweets as their induced redundancy may cause hindrance in Twitter analysis
- removal of '@' and words connected with it to smoothen the analysis
- removal of special characters including parenthesis and braces for the comfort of human experts.

5.3 Unicode language classification

The Unicode detection analysis is used for classifying the tweets to their respective scripts. We considered top three mostly communicated language in Indian Twitter, English, Hindi and Bengali for the purpose of analysis. Each script has its own unicode range, so comparing the unicode values of the tweets with those in any known range, scripts detection is performed which makes the filtering of the tweets easier for the system performance.

5.4 Ensembled approach

The hybrid approach includes a combination of machine learning tools and techniques to overcome the above problem as specified in Table-1. This approach comes into existence only when the Unicode detection results show the language detected as English. If English, then it will be further checked by the hybrid-based approach. The hybrid approach is based on supervised learning of Machine learning, as in this case, the training is provided with certain input data along with the corresponding output data, and the output of the classification result is known, that may be either English, Hindi or Bengali.

The following algorithm explains the steps involved in language identification of tweets in the Indic language by the ensembled algorithm. After the initial process of tweets collection, first tweets are classified using Unicode detection algorithm. Tweets with the detection result as English, are further checked using hybrid approach, i.e., ensemble of naive Bayes and SVM. If the detection result shows languages of the tweets as either Hindi or Bengali, they are further transliterated to their respective scripts. Once all the original scripts of the tweets are obtained, translation phase is executed. Finally, we have all the tweets transformed into English for being clustered into relative groups for further mining operations.

Classifier	*Predicted language*	*Precision of prediction*
SVM	X1	Y1
Naive Bayes	X2	Y2

Algorithm 1 Ensembled algorithm

```
1:  procedure ENSEMBLED(tweet)                          //Classifying tweets
2:      for all tweets t do
3:          if SVM(t)! = NULL then
4:              if SVM(X1) == Bayes(X2) then
5:                  L ← SVM(X1)
6:              else
7:                  if SVM(Y1) >= Bayes(Y2) then
8:                      L ← SVM(X1)
9:                  else
10:                     L ← Bayes(X2)
11:         else
12:             L ← Bayes(X2)
13:     return L                                        //Detected Language
```

5.5 Transliteration

Transliteration is the process of converting a word to its phonetic (similar sound) equivalent in another language. Our system needs transliteration for converting regional languages written in Roman scripts to their original script. Google Input Tools provide the best transliteration among all the other transliteration system available. Google Input Tools works by transliterating single word at a time that has been entered in the Transliteration Text Area. The word must be followed either by a 'Space', 'Enter' or 'Tab' key for the conversion to occur each time in the Online Google Input Tools.

Though the accuracy of Google Input Tools is pretty high but no predefined API is supplied by Google for transliteration except for an obsolete Google Transliteration API which was of no use as per our requirements. For that reason, we have developed an API which can use the Online Google Input Tools and makes transliteration for our system as per our need. The API is developed in Javascript with modules from 'NodeJS' and 'npm' for automation of the transliteration system. A robot module (provided by NodeJS) is used namely 'robot.js' for making the system automatic as a huge number of tweets are needed to be transliterated in a very short amount of time. The robot module is made to work by:

- taking input as sentence from the system and sending one word at a time to the Online Google Input Tools
- press the space key for the transliteration to occur and waits for the transliteration to complete
- after completion the transliterated word is stacked one after another until all the words of the sentence are transliterated
- at last the stacked transliterated words are appended to form a sentence and returned to the system.

For making this process easier for the frontend system we have created a function for our API

gTrans(sentence,lang,toLang)

This function consists of three parameters. Definition for the parameters is given below:

- sentence – this parameter accepts sentence as string for the system
- lang – this parameter takes the script language for the sentence which is set to 'English' for our system
- toLang – this parameter is the original language of the tweet (this field is populated by the language that is detected by the ensembled language classifier in the previous stage)

So, by using our simple function and API we can easily implement the Google Transliterate in our system. The transliteration algorithm is described below:

Algorithm 2 Transliteration algorithm

```
1: procedure ENSEMBLED(tweet)                                  // Transliteration of tweets
2:     for all tweets t do
3:         var sentence = tweets(t).text;
4:         var lang = "EN"                                          // lang field is English
5:         var toLang = tweets(t).lang;                  // toLang field detected language
6:         var doc = gTrans(sentence, lang, toLang);                  // defined function
7:     return doc          // the doc variable containing the transliterated sentence is returned
```

5.6 Translation

Translation phase will be executed under two situations. Firstly, if the language detected by the Unicode detection algorithm is not English, i.e., either Hindi or Bengali, means the script is actually in these languages and secondly, after the transliteration phase, when the scripts will be converted into their actual native languages, i.e., either Hindi or Bengali. Google provides open source translation API, and as it is mostly used for this reason we used Google API to translate the scripts in different languages, finally into English for further mining.

Table 5 shows the practical implementations of the above steps over the raw tweets, how the tweets are being practically transformed into different form after execution of each step.

5.7 Terminologies

In the context of a country like India, 'script vs language' problem for language identification is always prevailing. For example, 'ACCHE DIN AA GAYE' is a text in Roman script, but the context of the text can be evaluated in Hindi language. Same for 'AMI BHALO ACHI' where alphabets are in Roman scripts but the language used is Bengali. At times, the tweets are itself in the Indic scripts which calls for a different solution and otherwise tweets written in Roman but actually represents other language.

The proficiency of language classification is minimised to a great extent due to the improper identification of the intended languages which results in erroneous evaluation of all the further steps in the system. The scope of our work covers two prime languages of India along with English to process, evaluate and extract twitter texts. The tweets are first classified into distinct groups of scripts S = Rm, Hn, Bn through the UTF-16 character encoding. Then the tweets in Roman(Rm) are further analysed for their context evaluation and are categorised as per their intended language L = Eng, Hin, Ben.

The test cases that we consider in our work are as follows: SL = (Rm_Eng), (Rm_Hin), (Rm_Ben), (Dv_Hin), (Bn_Ben).

Table 5 Stage-wise system behaviour

Roman_Indic tweets	*Detected_lang*	*Transliteration*	*Translation*
ami valo achi tumi kemon acho	Bengali	আমি ভালো আছি তুমি কেমন আছো	I'm good how are you
kichu korar nai ar thamiye lav ki	Bengali	কিছু করার নাই আর থামিয়ে লাভ কি	There is no gain, and what to do to stop
aap aa chuki hai madam	Hindi	आप आ चुकी है मैडम	You've arrived ma'am
Jab sath nhi de sakte To jhutti umeeden bhi mat dilaya kro	Hindi	जब साथ नहीं दे सकते तो झुटटी उम्मीदें भी मत दिलाया करो	Do not give false hopes when you cannot accompany them

Table 6 Classifiers accuracy analysis

Algorithms	*Training*	*Rm_Hin*	*Rm_Ben*	*Rm_Eng*
Naïve Bayes classifier	10	86.83	75.83	96
	30	89	77.98	96.5
	50	96.66	80.78	96.57
	70	96.16	83.67	97.5
	75	96.33	88.33	97.83
Support vector machine	10	68.16	50.83	62
	30	75	67.33	69.33
	50	85	72.56	77.16
	70	89.18	79.03	82.03
	75	89.83	79.67	83.20
Ensembled algorithm	10	91.5	80	96.66
	30	96	82.5	97
	50	97	84.74	98.5
	70	97.70	88.33	99
	75	97	90.16	99.15

5.8 Comparison of algorithms

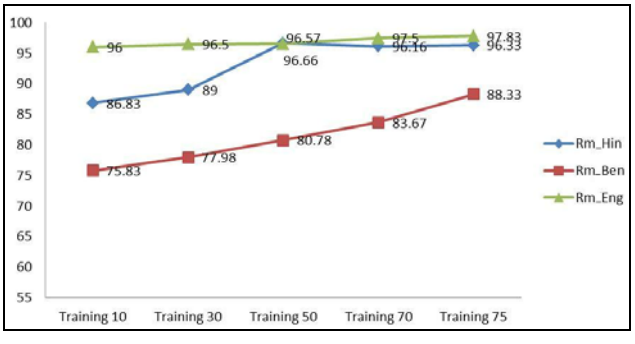
The tweets from the group of SL1, where SL1 = (Dv_Hin), (Bn_Ben), are directly sent to the translation phase for opinion mining and those from SL2, where SL2 = (Rm_Eng), (Rm_Hin), (Rm_Ben) are passed to the transliteration phase.

In the classification phase, three classification algorithms viz. naive Bayes, support vector machine and ensembled learning are implemented on each of the elements of the SL2 group. A rigorous learning period consisting of 10, 30, 50, 70 and 75 training data sets for each language is conducted for 3,000 tweets divided into equal partitions of 600 tweets for each of the SL2 elements applying all the three classification algorithms. The results found are encouraging for the implementation of the ensembled learning algorithm as it outperforms the other two algorithms by a significant margin. The performance matrix for the three classification algorithms on the elements of SL2 are shown in Table 6.

5.9 Graphical analysis

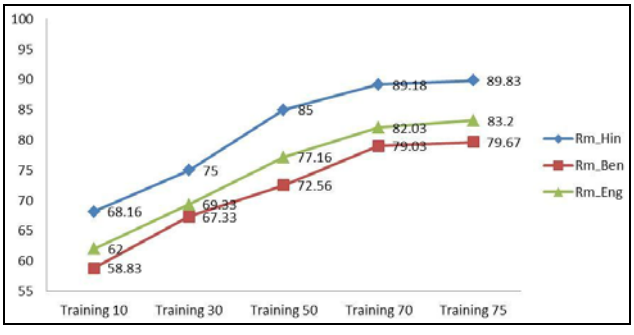
The analysis of the above three classifiers revealed very interesting outcomes for the research. Naive Bayes classifier, being one of the most primitive classifiers, is quite capable of accurately classifying English tweets from the very beginning with a little training but when the Indic languages are concerned naive Bayes cannot perform in the same manner as shown in Figure 3. Yet the remarkable feature of this classifier is that one of the Indic languages picks up the pace after a little bit of training and provides a steady progress for the other Indic languages.

Figure 3 Naive Bayes classifier (see online version for colours)



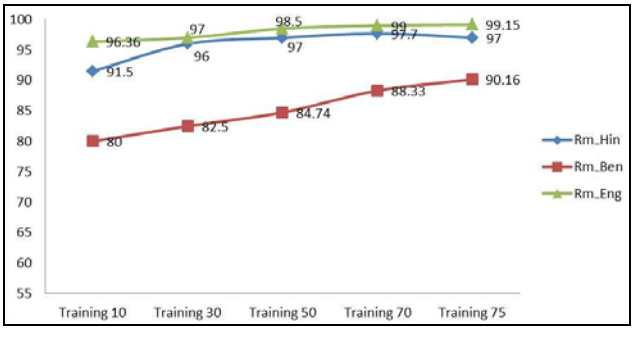
The second classifier in our consideration, i.e., support vector machine, shows a little better classification of the Indic languages vis-à-vis the English language, although the initial learning rate is much slower when compared to naive Bayes classifier. The SVM classifier used in this paper uses the 3-gram of characters to aid in the specification of the classifier. The ability of gradual and steady learning for all of the considered languages by the classifier is the key feature why we considered this as one of our chief components.

Figure 4 SVM classifier (see online version for colours)



Finally, when all the pros and cons of the two most powerful classification algorithms were evaluated, we designed and featured the ensembled algorithm after combining the positive features of both the classifiers and found that the accuracy of the system from the very beginning is quite high as well as the learning rate is also steep. This classifier gives higher accuracy from the beginning, like the naive Bayes classifier and also shows a steady pace of learning just like the support vector machine. It proved to be a major step in improved classification of the Indic languages.

Figure 5 Ensembled classifier (see online version for colours)



Once the classification of tweets from SL2 is over, they are then kept in separate databases for the ease of tagging them in the transliteration phase where they are transliterated to their respective scripts. The process is carried out by a robot programming module which transliterates the tweets one by one and then appends the transliterated (Rm_Hin), (Rm_Ben) to the elements (Hn_Hin), (Bn_Ben) of the SL1 group. Thus, SL1 now contains all the tweets needed to be translated and is forwarded to the translation phase. Hence, all the tweets get in the final form of (Rm_Eng) which is the required format for feeding to the topic clustering phase.

5.10 System performance

Table 7 shows the overall performance of the proposed system till finally translating tweets into English, over different corpus acquired. In this case the Rm_Eng is not considered, as it means English language tweets in Roman script and after the classifier detect it as Rm_Eng, which will be directly send to the final collection, without undergoing the phases described above.

Table 7 Overall system performance

	Rm_Hin	*Rm_Ben*	*Dv_Hin*	*Bn_Ben*
Tweets	3,000	3,000	3,000	3,000
True positive	2,819	2,400	2,824	2,600
False negative	181	600	176	3,000
TPR	93.96	80	94.13	86.67

Notes: True positive: number of tweets correctly detected as belong to the considered class (language).
False negative: number of tweets falsely detected as different class (language).
True positive rate (TPR): proportion of positives that are correctly identified.

5.11 Topic clustering

In the topic clustering phase, as most of the tweets will be of unknown nature, hence unsupervised approach has to be followed. Moreover, tweets have the ability to reflect more than one sense of feeling or pertaining to any single field. Indian Twitter when properly analysed can reveal multi-faced information from a handful of correctly analysed tweets. Hence, the generative statistical model to be used in the clustering phase must have the feature of analysing a single tweet in different senses. Latent Dirichlet allocation (LDA) has the desired features as discussed so far and hence our research employed LDA modelling for clustering the tweets into relevant clusters. On receiving all the tweets from the database, the LDA clusters all the tweets into clusters as it sees fit and afterward the list of generated clusters are labelled according to the most probable generic name.

Figure 6 Working of LDA (see online version for colours)

When the LDA completes its modelling, the clustered tweets are analysed to best fit into the classes previously decided. This is done for filtering the relevant tweets required to analyse at a particular instance. The end user is having the privilege to glance at the clustered topic generated after live streaming of tweets. Once the clusters of interest are chosen by the end-user out of all the available clusters, the contents of those clusters become available for inspection.

6 Conclusions and future scope

Online data retrieved from social media such as Twitter has offered immense potential for capturing valuable information that can be used to assess and solve many important real-life problems. But the effectiveness of this information mining application depends on the accuracy in interpreting the language being used. It becomes really difficult when users use different languages and/or different scripts.

In the Indian scenario, most of the tweets are erroneously classified as foreign languages and hence the evaluation and mining operations fail miserably. Our effort has defined a better way of identifying and classifying Indic tweets irrespective of the script and the language through the usage of an ensembled machine learning algorithm. The enhanced learning improved the later stages of analysis and is very helpful for efficient clustering of the relevant topics which otherwise would be lost without a competent classification mechanism.

In this paper, we considered only Twitter data. But the concept can easily be applied to all types of social networks such as Facebook, etc. We also restricted our focus on the Hindi, Bengali and English languages with either intrinsic or Roman script. But our approach can be scaled to any language and any script provided the corresponding transliterator and translator are available.

References

90% of New Net Users Non-English [online] http://timesofindia.indiatimes.com/business/india-business/90-of-new-net-users-non-english/articleshow/58371769.cms (accessed 4th May 2017).

Banerjee, S., Kuila, A., Roy, A., Naskar, S.K., Rosso, P. and Bandyopadhyay, S. (2015) 'A hybrid approach for transliterated word-level language identification: CRF with post-processing heuristics', *Proceedings of the Forum for Information Retrieval Evaluation*, pp.54–59, ISBN 978-1-4503-3755-7.

Barman, U., Das, A., Wagner, J. and Foster, J. (2014) 'Code-mixing: a challenge for language identification in the language of social media', *Proceedings of the First Workshop on Computational Approaches to Code-Switching*.

Bhargava, R., Sharma, R. and Sharma, S. (2016) 'Sentiment analysis for mixed script Indic sentences', *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp.524–529.

Cardoso, D., Miguel, P. and Roy, A. (2016) 'Language identification for social media: short messages and transliteration', *Proceedings of the 25th International Conference Companion on World Wide Web*, pp.611–614, ISBN 978-1-4503-4144-8.

Carter, S., Weerkamp, W. and Tsagkias, M. (2013) 'Microblog language identification: overcoming the limitations of short, unedited and idiomatic text', *Lang. Resour. Eval.*, Vol. 47, No. 1, pp.195–215, ISSN 1574-020X.

Chittaranjan, G., Vyas, Y., Bali, K. and Choudhury, M. (2014) 'Word-level language identification using CRF: code-switching shared task report of MSR India System', *Proceedings of the First Workshop on Computational Approaches to Code Switching, EMNLP-2014*, pp.73–79.

Dutta, S., Saha, T., Banerjee, S. and Naskar, S.K. (2015) 'Text normalization in code-mixed social media text', *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*, pp.378–382.

Ertel, W. (2011) 'Machine learning and data mining', *Introduction to Artificial Intelligence*, pp.161–220, Springer, London, ISBN 978-0-85729-299-5.

Gold, E.M. (1967) 'Language identification in the limit', *Information and Control*, Vol. 10, No. 5, pp.447–474.

Gottron, T. and Lipka, N. (2010) 'A comparison of language identification approaches on short, query-style texts', *Proceedings of the 32nd European Conference on Advances in Information Retrieval*, pp.611–614, ISBN 3-642-12274-4, 978-3-642-12274-3.

Hong, L. and Davison, B.D. (2010) 'Empirical study of topic modeling in Twitter', *Proceedings of the First Workshop on Social Media Analytics*, pp.80–88, ISBN 978-1-4503-0217-3.

Hong, L., Convertino, G. and Chi, E.H. (2011) 'Language matters in Twitter: a large scale study', *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media ICW SM 11*, pp.518–521.

King, B. and Abney, S. (2013) 'Labeling the languages of words in mixed-language documents using weakly supervised methods', *Proceedings of NAACL-HLT 2013*, pp.1110–1119.

Laboreiro, G., Bošnjak, M., Sarmento, L., Rodrigues, E.M. and Oliveira, E. (2013) 'Determining language variant in microblog messages', *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pp.902–907, ISBN 978-1-4503-1656-9.

Lin, J. and Ryaboy, D. (2013) 'Scaling big data mining infrastructure: the Twitter experience', *SIGKDD Explor. Newsl.*, Vol. 14, No. 2, pp.6–19, ISSN 1931-0145.

Lui, M. and Baldwin, T. (2011) 'Langid.Py: an off-the-shelf language identification tool', *Proceedings of the ACL 2012 System Demonstrations*, pp.25–30.

Lui, M. and Baldwin, T. (2014) 'Accurate language identification of Twitter messages', *Proceedings of the 5th Workshop on Language Analytics for Social Media (LASM) @EACL 2014*, pp.17–25.

Mehrotra, R., Sanner, S., Buntine, W. and Xie, L. (2013) 'Improving LDA topic models for microblogs via tweet pooling and automatic labeling', *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.889–892, ISBN 978-1-4503-2034-4.

New Research Reveals Global Social Media Use Increased by 21 Percent in 2016 [online] http://www.marketwired.com/press-release/new-research-reveals-global-social-media-use-increased-by-21-percent-in-2016-2190591.htm (accessed 6th May 2017).

Number of Twitter Users in India from 2012 to 2019 (in Millions) [online] https://www.statista.com/statistics/381832/twitter-users-india/ (accessed 6th May 2017).

Sharma, S., Agrawal, J., Agarwal, S. and Sharma, S. (2013) 'Machine learning techniques for data mining: a survey', *2013 IEEE International Conference on Computational Intelligence and Computing Research*, pp.1–6.

Sharma, S., Srinivas, P. and Balabantaray, R.C. (2015) 'Text normalization of code mix and sentiment analysis', *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp.1468–1473.

Singh, A.K. and Goyal, P. (2014) 'A language identification method applied to Twitter data', *TweetLID @SEPLN 2014*, pp.26–29.

The Growth of Social Media: From Passing Trend to International Obsession [Infographic] [online] http://www.adweek.com/digital/the-growth-of-social-media-from-trend-to-obsession-infographic/ (accessed 4th May 2017).

Truica, C.O., Velcin, J. and Boicea, A. (2015) 'Automatic language identification for romance languages using stop words and diacritics', *2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp.243–246.

Twitter Usage Statistics [online] http://www.internetlivestats.com/twitter-statistics/ (accessed 4th May 2017).

Williams, J. and Dagli, C.K. (2017) 'Twitter language identification of similar languages and dialects without ground truth', *EACL-2017 VarDial Workshop*, pp.73–83.

Year-on-year Twitter User Growth Rate in India from 2013 to 2019 [online] https://www.statista.com/statistics/303691/twitters-annual-growth-rate-in-india/ (accessed 4th May 2017).

Zou, L. and Song, W.W. (2016) 'LDA-TM: a two-step approach to Twitter topic data clustering', *2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pp.342–347.