SENTIMENT ANALYSIS OF MIXED CODE FOR THE TRANSLITERATED HINDI AND MARATHI TEXTS

Mohammed Arshad Ansari and Sharvari Govilkar

Department of Information Technology, Pillai College of Engineering, New Panvel, Navi Mumbai, Maharashtra, India 410206
Department of Computer Engineering, Pillai College of Engineering, New Panvel, Navi Mumbai, Maharashtra, India 410206

ABSTRACT

The evolution of information Technology has led to the collection of large amount of data, the volume of which has increased to the extent that in last two years the data produced is greater than all the data ever recorded in human history. This has necessitated use of machines to understand, interpret and apply data, without manual involvement. A lot of these texts are available in transliterated code-mixed form, which due to the complexity are very difficult to analyze. The work already performed in this area is progressing at great pace and this work hopes to be a way to push that work further. The designed system is an effort which classifies Hindi as well as Marathi text transliterated (Romanized) documents automatically using supervised learning methods (KNN), Naïve Bayes and Support Vector Machine (SVM)) and ontology based classification; and results are compared to in order to decide which methodology is better suited in handling of these documents. As we will see, the plain machine learning algorithm applications are just as or in many cases are much better in performance than the more analytical approach.

GENERAL TERMS

Natural Language Processing, K-NN, K Nearest Neighbor, Support Vector Machine, Naïve Bayes, Ontology, Precision, Recall, F-measure, Confusion Matrix, Random Forest.

KEYWORDS

Code-mix, Hindi, Marathi, Transliteration, Sentiment Analysis.

1. INTRODUCTION

Sentiment Analysis has become the area of deep research due to necessity wrought about by the advent of social media tools such as Twitter, Facebook, WhatsApp, etc. These social media tools have made their mark when they became the primary tool for the collaboration by masses during various public movements of great import; such as, Arab Spring, Occupy Wall Street, to name the few. It has come to the attention of not just the content marketing industry, but also the Governments across the world that these social tools can be a great tool for the benefit of mankind or threat to national security or even a great profit generation system. The basic idea behind sentiment analysis is to tap the nerves of the masses, by identifying and mapping the direction of opinions that lead to various trends. These trends in turn can be tapped to shape,

manipulate or even understand the mode of people's reaction to the events around the world. Especially during the time, when the expression of emotions through social media is not just available but has also permeated the lives of people and with new generation completely depending on such tools. Therefore, the research in this area is not just booming but is also one of the areas that is being tread carefully, due to the implications as have been suggested above.

If we dive deeper in the research about Sentiment Analysis, there is one area which has been the major bane in the field without an appropriate solution that has been accepted as the answer by the researchers in the field. This area is marked by the advent of code-mix text, which is the result of multi-lingual, multi-cultural and multi-racial society that we find ourselves part of. Today, the world over, people communicate using not just one language, but rather with a mixture of more than one language. These mixtures of languages usually have got English as one component and the mother tongue of the speaker. In Mumbai, where the people speak all sorts of language; there are two combinations that stand out, in terms of usage and coverage. These combinations are English-Hindi and English-Marathi. Another major combination is Hindi-Marathi or English-Hindi-Marathi, however; they are not part of the scope of this research. This research focuses on the first two mentioned combinations. It has been found that the people tend to use Latin alphabets to phonetically type the non-English words and sometimes, the phonetic spelling is also extended to English words, especially those, whose spelling is not so obvious.

Sentiment Analysis of such a plain text in English, Hindi or Marathi is already a solved problem, with improvements being done each day and new tools coming up to increase the accuracy. However, if the same is to be done for code-mix script then it becomes a much involving task with accuracies ranging from very low to somewhat medium. There is a growing need to inculcate the ability to analyze the mixed script, if the benefits mentioned in the previous paragraphs is to be achieved. And therein lies the real problem with great amount of work being done on the subject matter. In this research, we focus on the two combinations, as mentioned above, English-Hindi and English-Marathi, with the script being used is of Latin characters. The reason for the choice of these characters is due to the usage by the people for communication, including the author himself. There are a lot of areas that needs to be covered when performing sentiment analysis of code-mix script; such as, Language Identification, POS Tagging, Sentiment scores generation and training the machine learning algorithm to perform the classification. As part of the scope of this research we have identified the bottleneck which needs to be fixed to increase the accuracy of the system and in what is to follow we will explain the proposed system.

2. LITERATURE SURVEY

Although the research on sentiment analysis has entered mainstream production scenarios, there has been great strides done in the field from all facets of research perspectives. In this section, we shall look at what work has been already done to accomplish what is being presented in this dissertation.

Code-mixing techniques and related work starts decades ago with initial work by Gold [1] with the aim of language identification in the mixed code text, in which, the language structure is obtained by learning the structure of language from informant and/or the given text and it concludes that the rule detection by pure text alone does not work and is highly dependent on the information with preconfigured rules of the language to be identified. Annamalai's [2]

pioneering work in the research field of language identification opened the door for the Indian languages, by identifying that a lot of corpus and usages in the Indian subcontinent. Nigam, Lafferty and Mccallum [3] used maximum entropy classifier for text classification with about 80% accuracy.

A English language SentiWordNet was introduced by the work of Esuli and Sebastiani [4], which became of primary importance for all sense based lexical analysis work to be produced after that time. An approach for language transliteration was given by Chinnakotla and Damani [5] using the character-based sequence modelling by applying the techniques such as Conditional Random Field. Such transliteration accuracy produced by statistical techniques were thus improved by using the lexical analysis tools and were shown to have performed better by the work of Khapra and Bhattacharyy.[6] SVM and Naïve Bayes was used to classify and label millions of comments for sentiment polarity by Siersdorfer, Chelaru et al [7] in their work.

Joshi, Balamurali and Bhattacharyya [8] has compared three approaches for the sentiment analysis of Hindi text and found that HSWN performs better than Machine Translation approach but under-performs machine learning approach in 2010. The same group also went ahead and created a sentiment analyzer for twitter microblogs using the approaches mentioned above, in their work [9] that followed in year 2011. They proposed sentiment classification using emoticons-based detector as well as lexicon-based detector.

A text normalization technique based on handling of abbreviations, spelling errors, missing punctuations, slangs, word play, censor avoidance and emoticons were shown to be working by the work of Clark and Araki. [10] Elfardy and Diab [11] coined the term code-switching in text during a recent research on the subject and gave a method to identify the code-switch occurring in the texts.

Balamurali and Joshi [12] demonstrated that lexical analysis using sense from SentiWordNet has tendency to outperform the normal word-based analysis in their work and created a robust sentiment analysis system that harnesses that approach in another of their work in year 2011 as cited above. Bakliwal, Arora and Verma [13] developed subjective lexical resource using only wordnet and graph traversal algorithm for adverbs and adjectives. In one of the experiments the team figured out that machine translation-based approach under-performs much worse compared to language using sentiment training and also one of the results was development of Hindi SentiWordNet using linked wordnet analysis. Rana [14] used fuzzy logic membership function to determine the degree of polarity of the sentiment for a given POS tagged prepositions.

Karimi, Scholer and Turpin [15] proposed machine translation techniques for the purpose of transliteration in a survey and gave a suggestion for the phoneme-based approach for transliteration generation using bilingual corpus, by applying CRF based classification technique to do so. Dewaele [16] in a work suggested that there is strong correlation between the emotional content of the text and the switching of code, to the extent that there is a probable causality between the two. The major issue of identification of language of the code - mix script is yet bigger challenge that shall be answered here. Kundu and Chandra [17] suggested a statistical approach to automatically detect English words in Bengali + English (Benglish) text.
Gupta, Chaudhury and Bali [18] mined the transliteration pairs from the music lyrics of Bollywood songs, in English-Hindi pairs, for Fire'14 shared task, which became one of the

standard bilingual word replacement tool for this work too. Balamurali [19] suggested another way of improving the sentiment classification by introducing the word liking ability using WordNet based sense similarities. Mittal and Aggarwal [20] used negation and discourse analysis to further the cause of sentiment analysis to reach 80.21% accuracy. A remarkable effort that setup a standard in the area language identification and labelling was done by Gella, Sharma and Bali [21], using the supervised methods to classify the word and included multi-grams for feature modelling. The algorithms used were Max Entropy algorithm, Naïve Bayes and Decision tree with Max Entropy outperforming the others and Naïve Bayes being close to second. POS Tagging effort was undertaken by Vyas, Gella et al [22] to conclude that it benefits overall, although the current work here suggests otherwise. Popat, Bhattacharyya and Balamurali, in their work [23] used clustering methods to improve accuracy of sentiment classification by applying approaches such as sense based and cross lingual word clustering by word sense.

King and Abney [24] using a conditional random field model with weakly supervised learning model for token labelling with near 90 percent accuracy. Barman, Das et al [25] uses social media data for language identification in mixed script and concluded in favor of supervised learning against the dictionary-based approaches. Nagesh and Ravi [26] gave a way to perform language identification using multi class regression classifiers and was able to get nearly 54% accuracy. Pandey and Sharvari [27] applied HSWN along with negation discourse for sentiment analysis of Hindi language text corpora, with the accuracy of near 80%. Srinivas, Sharma and Balbantray [28] demonstrated that text normalization can be achieved using techniques such phonetics based, slang and spelling correction approaches in another limited work [29] they demonstrated application of sentiment analysis techniques on transliterated text by using bilingual dictionary methods and HSWN for sentiment score calculation with 80% accuracy being achieved. Marathi and Sanskrit word identification was achieved by another work by Kulkarni, Patil and Dhanokar [30] using the genetic algorithm.

3. PROPOSED METHODOLOGY

Initially, the proposed system was to contain steps directly pulled form usual steps in the art of sentiment analysis, such as POS tagging, etc, however; it was found out earlier that there were many problems associated with the methodology of sentiment analysis using the standard method that has proved so successful for single language corpus. Before moving on the proposed system we will look at the problems uncovered with the traditional approach of sentiment analysis.

Problems with the traditional approach to sentiment analysis

1. Language identification using dictionary lookup failed, because the words that people generally use, vary in their spellings and are highly contextual. There are so many commonly used words with multiple spelling variations such as like: “lik”, “lk”, etc. and other words “mein” being used as me and ma used for mother as well as “my” in colloquial language. These variations cause the words to not be identified correctly belong to any particular language, let alone correcting the spelling.

2. The other big problem was with POS tagging. This problem arises on two fronts, namely;

   a. The chunks cause the entire structure of the statement to be dismantled and POS tagging sees only the part of the system at a time, therefore making faulty judgements on the kind of tag to be applied on the word.
   b. Misidentified words throw the POS tagging way of the correct usage and results in completely incorrect tags to be applied.
3. The third and derivative problem is that due to misidentification of words and misapplication of tags, the sentiment wordnet (SWNs) become useless, since the wrong word in wrong SWN is being looked up, or the wrong POS tag value is being used to look it up.

All the above three problems combine to reduce results to less than 40% peak and 25% average accuracy of polarity identification, which is worse than training directly the whole statement without doing anything on a simple Naïve classifier and get the results around 52-55% accuracy. Although, the problems enlisted above are not completely removed from the proposed system, however; they have been mitigated to a large extent and therefore, a work was able to produce some good results with data that has not been used for the specific purpose of sentiment analysis. In this sense, this is a pioneering work to make sense of social data to an extent that it can be analyzed and a result thus obtained is on par with the state of the art systems.

The system that was adopted is simple and have been arrived at refining and removing the inconsequential aspects of the early proposed systems. The experiments were performed with all the data and it was realized after multiple improvements that the proposed architecture cannot be improved further without attacking the real problems of the domain. In the following section, we describe the steps.

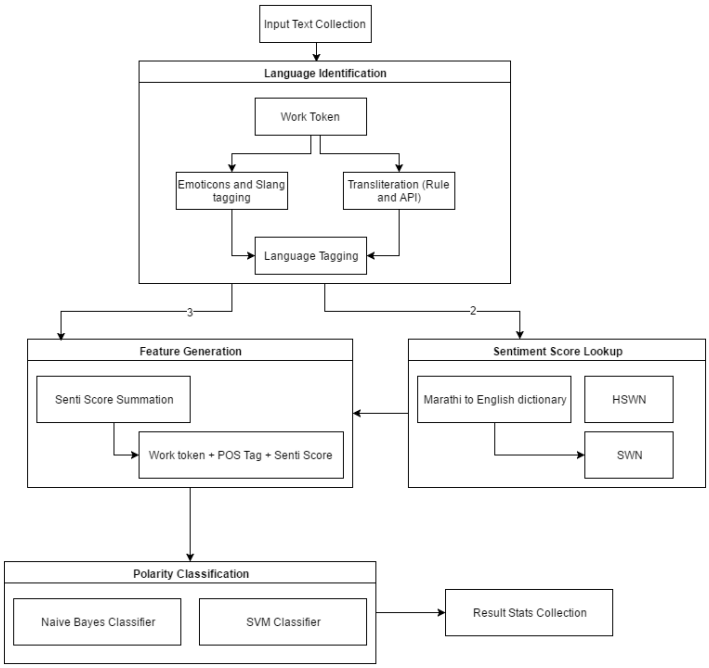

Figure 1: System Architecture

The proposed methodology consists of following phases:

1. Language Identification
2. Word Transliteration
3. Sentiment Scores Tagging
4. Feature Extraction
5. Supervised Learning Methods
6. Output as sentiments of Hindi/Marathi text documents

3.1 Language Identification

There are many ways to identify the word, regarding the language it belongs to. The below given list of steps are quite rudimentary compared to the actual code that went in to build the identifier. Emotions conveyed using emoticons are converted to words that can be used for further analysis. Since this step is difficult, as was seen earlier and algorithm to achieve this step is show below:

Algorithm: Language Identification
Steps:

a. Replace emoticons with words
b. Check if the word group is in slang dictionary then capture the sentiments for word from slang dictionary with their corresponding sentiments.
c. Transliterate the word and check if the word is in Hindi/Marathi Wordnet then capture the Hindi/Marathi word (see point 2 for transliteration) with association of the same work in English script.
d. Check in English dictionary for the existence of word and capture the work if it exists.

Example:

e. Lol, itna funny statement tha na who :)
f. lol, itna funny statement that na woh. smile
g. laugh out loud, itna funny tha na woh. Smile
h. Output

English: laugh, out, loud, funny, smile
Other (hindi): itna, tha, na, who
Slang: lol = laugh out loud (with slang based sentiment scores)

3.1.1 Emoticons and slang identification

One of the important steps of preprocessing is to identify usages of emoticons in the text. They are of high value to classify the sentiments and therefore require a separate to look up emoticons, slangs or abbreviated short forms with appropriate textual data which will be picked up later for analysis.

3.1.2 Dictionary lookup and Frequency based selection

The words thus obtained from previous steps are looked up in English as well as Hindi/Marathi dictionaries. Word frequencies are used to resolve ties when the words are found to be in

multiple dictionaries. Since the words used many a times are social media centric, this step hardly resolves much of the ambiguities from the documents, since a lot of words are not present or morphed phonetically in general during the daily usages in conversations.

3.2 Transliteration to native languages

After the word is identified to belong to a particular native language, then it is transliterated to the respective text, in our case, the Devanagari script. This transliteration process is a preparatory step for what is to follow. There is an available Sanscript translation engine listed in the references, which is used for the process of transliteration. However, it is completed also with the google transliteration API, in order to achieve greater accuracy in transliteration.

Algorithms: Transliterate Word

Steps:

a. Generate upto 5 ngrams of words collection sets and for each ngram set do the following:
b. Use the Sanscript tool to convert the English script work to Devanagari word
c. Use Google Language Transliteration API to translate and collect the collection of possibilities.
d. Use Hindi / Marathi Wordnet based dictionary to check for the existence of word and capture the synset id for later use for collecting senti scores.
e. For every word in ngram that was found, add it to the collection of word that will be need for sentiment scoring.

Example:

f. laugh out loud itna funny that na smile
g. laugh, out, loud, itna (इतना), funny, tha (था), na (ना), woh (वह), smile

3.3 Sentiment score tagging

The transliteration obtained from the previous step is used in this step to find out the sentiment scores associated with the given word from the SentiWordNet available with us for both English and Hindi language. The case for Marathi is a bit tricky, since there isn't a SentiWordNet available for it, however; this short coming is overcome with the help of bilingual dictionary to translate the Marathi word to Hindi word first and then use HSWN to arrive at the sentiment scores. These scores form the features with which the data is annotated.

Algorithms: Sentiment Score Lookup

Steps:

a. For each word in the ngram collection select the SentiWordnet based on the language identified
b. In case of Marathi language choose English sentiwordnet and use bilingual dictionary to choose corresponding Hindi word for Marathi word

c. Use All POS Tags for the word and lookup the sentiments associated with the word and average them.
d. If the word is not available in SWN then assign 0 values for all polarity values

3.4 Feature Generation

Feature Generation is a time-consuming step that combines with all the above steps and gives a result, which is used as an input for machine learning based classification. As can be seen in the example below, the feature consists of the language information and features such as word, word type, positive sentiment score and negative sentiment score. All this data can be transformed to various other format for the purpose of fitting it according to the requirement of the machine learning algorithm in use.

Algorithms: Feature Generation

Steps:

a. Create a list of to hold all features
b. For every ngram word set in the text do the following
c. Generate the feature given below as shown in example for each ngram phrase.
d. Remove the lesser size ngram if higher size ngram is present with the sentiments
e. Replace English word ngram with slang ngram if found
f. Order the ngrams according to the sentence structure
g. Add it to the list of features
h. Scores of all the words are summed up associated with the list
i. Return the list

Since a lot of N-Grams are calculated (upto 5) and each one is processed for generation of features, the time it takes to complete this step is pretty high. The output of this process is then fed to the classifier which is able to perform the sentiment classification of the given documents.

3.5 Classification of documents using supervised learning methods

Finally in this phase, we will get output as set of classified documents as per class label, which would be the sentiments of the document. Classification of documents are tested against two classification algorithms such as Naïve Bayes algorithm and Support Vector Machine (again using RBF and Linear SVM). As proposed in the original early proposal Conditional Random Field is not being used due to its very slow performance and not different from the above-mentioned algorithms in accuracy.

Algorithms: Classification Algorithm

Steps:

a. Create a percent list as shown [5, 10, 15, 25, 50, 75]
b. For every percent in the percent list:
c. Train that amount of input text
d. Test against the remaining percentage of input text
e. Collect the results accuracy for RBF SVM, Linear SVM and Naïve Bayes

f. Collect the maximum accuracy from all the algorithms and note it separately
g. Average the results and return the average result
h. Calculate the accuracy percentage

4. RESULTS AND DISCUSSION

4.1.1 Dataset used

Total 1200 Hindi and about 300 Marathi documents from social media were used, which includes data from: Chats, Tweets, YouTube comments.

4.1.2 Performance Measures

Many performance measures are usually applied to evaluate they text classifiers. The designed system adopts standard performance measures such as precision (False Positive), recall (False Negative) and f-measure.

The results are generated by varying the number of documents used for the training and the accuracy is matched to find the peak that lies between under and over training of the classifier.

Table 1: Result of F-Measure for Multiple Classification Techniques for Hindi Language

Train Percent	Max F1 Score	SVM (linear) F1 Score	SVM (RBF) F1 Score	Naive Bayes F1 Score
4	0.54	0.43	0.28	0.46
8	0.54	0.47	0.22	0.49
12	0.56	0.53	0.38	0.55
16	0.58	0.56	0.33	0.56
20	0.59	0.55	0.39	0.56
24	0.58	0.57	0.31	0.55
28	0.59	0.57	0.31	0.56
32	0.61	0.55	0.39	0.57
36	0.61	0.6	0.31	0.57
40	0.61	0.6	0.33	0.59
44	0.61	0.57	0.28	0.57
48	0.61	0.6	0.31	0.6

The Hindi dataset trained with varying the percentage of data to be used for training is shown in the above table and it can be seen that F1 Score for NB and Linear SVM is higher than that of F1 Score of RBF based SVM. We stopped just before reaching 50% of the data for training, because of the problem of overfitting that becomes evident and because, it is as bad as mapping all the data one on one to the sentiments and can produce misleading results. Similarly, below table shows the F1 Score of Marathi Dataset and it is very clearly visible that the same two algorithms are performing as in the case of Hindi Dataset.

Table 2: Result of F-Measure for Multiple Classification Techniques for Hindi Language

Train Percent	Max F1 Score	SVM (linear) F1 Score	SVM (RBF) F1 Score	Naive Bayes F1 Score
24	0.63	0.59	0.58	0.61
28	0.67	0.64	0.51	0.54
32	0.61	0.56	0.49	0.6
36	0.64	0.61	0.48	0.65
40	0.72	0.6	0.61	0.68
44	0.7	0.55	0.47	0.69
48	0.69	0.59	0.52	0.67
4	0.65	0.6	0.59	0.51
8	0.57	0.44	0.12	0.36
12	0.57	0.53	0.51	0.47
16	0.58	0.6	0.58	0.48
20	0.59	0.59	0.52	0.46

Following is the chart that compares all the algorithms we have seen so far to be able to better view the differences and performance capabilities in each algorithm in comparison to others.

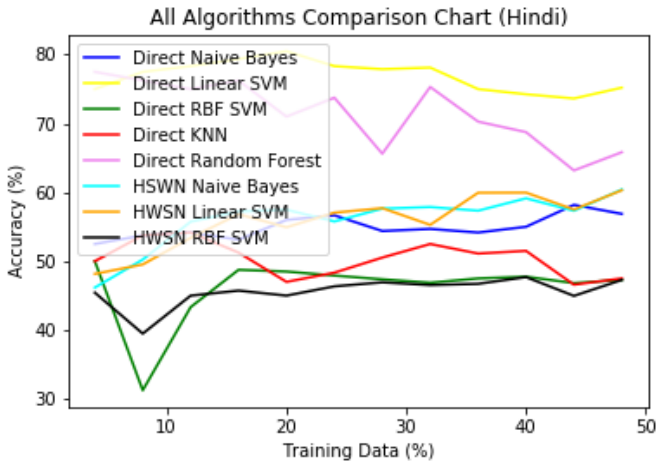

Figure 2: Graphical representation of accuracies obtained for Hindi language

The highest performing algorithm are direct linear SVM and direct Random Forest algorithms, followed by lexical analysis based Naïve Bayes algorithm and lexical analysis based Linear SVM for Hindi language. The worst performing algorithm was lexical analysis based RBF SVM and direct RBF SVM. The below chart represents the same chart for Marathi language and has pretty stark contradictions from Hindi language. Here, the direct algorithms such as Random Forest, Linear SVM and KNN to some extent, makes for the best performing algorithm, while at the same time, lexical analysis based Naïve Bayes algorithm performs worst on an average. It is very interesting to see that the Marathi language is able to reach the accuracy levels up to 90% with

somewhat consistency, unlike the Hindi language, which rarely reaches 80% accuracy and barely stays above 70% accuracy.

The current system deals with the social media data and therefore, is bound to have not only inconsistencies in word spellings but also contains ambiguity in the usage of such words. The data collected was from fire 2013 as well as YouTube and Twitter. This data was manually annotated, to train the machine learning algorithm. The tools provided by Scikit-learn are quite fulfilling as far as the need for testing goes. The entire system is focused on research level usage and therefore has many tools available as built-in to save development time and make testing easier as well as transparent. As we have seen, we were able to leverage the metric calculation of the given system to calculate F1 scores of the runs and thereby achieve higher level of precision. Validation of the tests were directly performed by using the inbuilt tools for machine learning as well as performance metric calculation. We have already seen the results collected via such system, come to use in so many ways that the entire task becomes not only easier but also trackable. This is the result of great community around Scikit-learn to gain understanding as well as better approaches to the problem.

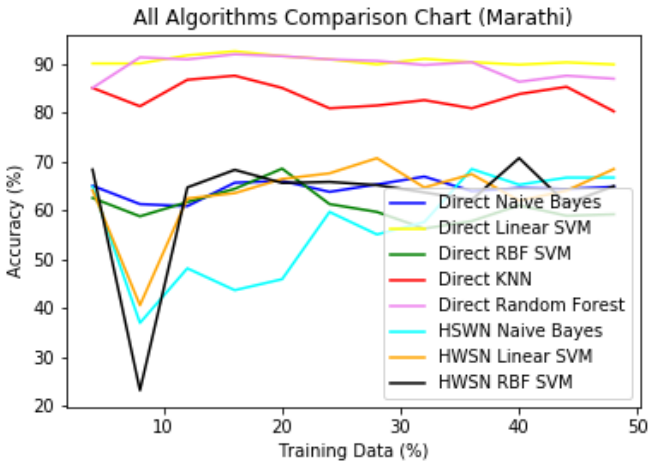

Figure 3: Graphical representation of accuracies obtained for Marathi language

5. CONCLUSION

The current work sheds a light on the combined applications of sentiment analysis, social media text normalization and code-mixed script, with the result in favor of the line of research as well as the possible returns from the same. It was one of the objectives to ensure that the field of sentiment analysis get applied on a more social context as well as on the languages that are used in day to day communication of general people. To that effect, the work has provided a very open, flexible and enhance able approach that can be considered as a successful direction. The source code as well as the data used for this project has been put on the Author's GitHub profile as open source repository for future use by those who will extend this research. The work also resulted in creation of a new open source project for Marathi Wordnet in python and the source code of that has been release on the GitHub website (github.com/arshadansari27).

From what was gathered with this experiment, it was clear that the real factor behind the accuracy levels of the system are highly dependent on social media platform context for communication. Without this, there will remain no further scope in this work, since the translation is already easy across language and sentiment analysis of this information is highly dependent on slang oriented, non-official usage of the language. The most improvement needed is in thus, language identification. Currently, the language identification step is highly dependent of the usage of wordnet and text normalization. A great amount of work is being done in normalizing the social media text for English language. However, same amount of work is needed to capture the similarly important trends of colloquial usage of regional languages. This corpus will enable the language identification techniques to not just solely depend on dictionary words but also day to day used words with their very unexpected sentimental values. There seems to be a huge scope for the development of the SentiWordNet as well for Marathi language and improvement of the existing HSWN. This improvement can come by adding more and more colloquial words for sentiment scores.

Another major area of improvement was grammar management of mixed code script, since the POS tagging goes out of the window for mixed scripts, as they break sentence structure, while moving from one language to next in the mid-sentence. It was also pointed out by one of the authors of the paper on sentiment analysis, that the code mix results from code switching, which is some characteristics of presence of emotional investment in the dialog and therefore, the need for focusing on the switching aspect to put a needle on the sentimental value to such switches. This work resulted in the creation of Marathi wordnet in python on the author's GitHub profile and therefore, as required by the project, can benefit from wide range of development efforts in related area. The future work remaining on this project is to improve HSWN and have a separate MSWN for Marathi language. As well as, the collection of corpus for regional language slangs to get a fix on the sentiment values of such usage of regional languages. Another key area of improvement identified, besides language identification, is in POS tagging of mixed script. This would add a very important tool in the quest of sentiment analysis of transliterated regional languages in English Script.

As it was established in this work that the possibility of using machine learning algorithms directly has a very advantageous position at the moment, there is still a great scope for improvement of the proposed system using this knowledge. For instance, it is possible to not use rule based language identification as was done in this work and use one of the machine learning techniques to perform language identification of the words. That is one step which would surely improve the performance and is the direction we are going to take moving forward.

6. ACKNOWLEDGMENTS

We would like to extent our gratitude towards the Information Technology department of Pillai College of Engineering New Panvel for giving us the opportunity to conduct the research. This research paper wouldn't have been possible without the efforts of our principal.

REFERENCES

[1] E. M. Gold and T. R. Corporation, "Language identification in the limit," *Inf. Control*, vol. 10, no. 5, pp. 447–474, May 1967.

[2] E. Annamalai, "The anglicized Indian languages: A case of code mixing," *Int. J. Dravidian*

Linguist., vol. 7, no. 2, pp. 239–247, 1978.

[3] K. Nigam, J. Lafferty, and A. Mccallum, "Using Maximum Entropy for Text Classification," *IJCAI-99 Work. Mach. Learn. Inf. Filter.*, pp. 61–67, 1999.

[4] A. Esuli, F. Sebastiani, and V. G. Moruzzi, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining," *Proc. Lr. 2006*, vol. 0, pp. 417–422, 2006.

[5] M. K. Chinnakotla and O. P. Damani, "Character Sequence Modeling for Transliteration," 2009.

[6] M. M. Khapra, "Domain Specific Iterative Word Sense Disambiguation in a Multilingual Setting," *Most*, 2008.

[7] S. Siersdorfer, S. Chelaru, W. Nejdl, and J. San Pedro, "How useful are your comments?," *Proc. 19th Int. Conf. World Wide Web*, vol. 15, pp. 891–900, 2010.

[8] A. Joshi, B. A. R, and P. Bhattacharyya, "A Fall-back Strategy for Sentiment Analysis in Hindi: a Case Study," no. October 2015, 2010.

[9] a R. Balamurali, A. Joshi, and P. Bhattacharyya, "Harnessing WordNet Senses for Supervised Sentiment Classification," *Proc. Conf. Empir. Methods Nat. Lang. Process.*, no. 2002, pp. 1081–1091, 2011.

[10] E. Clark and K. Araki, "Text normalization in social media: Progress, problems and applications for a pre-processing system of casual English," *Procedia - Soc. Behav. Sci.*, vol. 27, no. Pacling, pp. 2–11, 2011.

[11] H. Elfardy and M. T. Diab, "Token Level Identification of Linguistic Code Switching.," in *COLING (Posters)*, 2012, pp. 287–296.

[12] B. A. R, A. Joshi, and P. Bhattacharyya, "Robust Sense-based Sentiment Classification," *Proc. Work. Comput. Approaches to Subj. Sentim. Anal. WASSA*, no. October, pp. 132–138, 2011.

[13] A. Bakliwal, P. Arora, and V. Varma, "Hindi Subjective Lexicon: A Lexical Resource for Hindi Polarity Classification," *The eighth international conference on Language Resources and Evaluation*, no. May. pp. 1189–1196, 2012.

[14] S. Rana, "Sentiment Analysis for Hindi Text using Fuzzy Logic," *Indian J. Appl. Res.*, vol. 4, no. 8, p. 16, 2014.

[15] S. Karimi, F. Scholer, and A. Turpin, "Machine transliteration survey," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 1–46, Apr. 2011.

[16] J.-M. Dewaele, "Emotions in multiple languages," *International Journal of Multilingualism*, vol. 9, no. 1. pp. 129–130, 2012.

[17] B. Kundu and S. Chandra, "Automatic detection of English words in Benglish text: A statistical approach," in *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*, 2012, pp. 1–4.

[18] K. Gupta, M. Choudhury, and K. Bali, "Mining Hindi-English Transliteration Pairs from Online Hindi Lyrics," *Proc. Eighth Int. Conf. Lang. Resour. Eval.*, pp. 2459–2465, 2012.

[19] B. A. R, "Cross-lingual sentiment analysis for Indian languages using linked wordnets," *Proc. COLING 2012*, vol. 1, no. December 2012, pp. 73–82, 2012.

[20] N. Mittal and B. Agarwal, "Sentiment Analysis of Hindi Review based on Negation and Discourse Relation," in *Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 57–62.

[21] S. Gella, J. Sharma, and K. Bali, "Query word labeling and Back Transliteration for Indian Languages: Shared task system description," 2013.

[22] Y. Vyas, S. Gella, J. Sharma, K. Bali, and M. Choudhury, "POS Tagging of English-Hindi Code-Mixed Social Media Content," *"Proceedings Conf. Empir. Methods Nat. Lang. Process.*, pp. 974–979, 2014.

[23] K. Popat, B. A. R., P. Bhattacharyya, and G. Haffari, "The Haves and the Have-Nots : Leveraging Unlabelled Corpora for Sentiment Analysis," *Proc. 51st Annu. Meet. Assoc. Comput. Linguist.*, no. October 2015, pp. 412–422, 2013.

[24] B. King and S. Abney, "Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods," *Proc. NAACL-HLT*, no. June, pp. 1110–1119, 2013.

[25] U. Barman, A. Das, J. Wagner, and J. Foster, "Code Mixing: A Challenge for Language

Identification in the Language of Social Media," in *Proceedings of the First Workshop on Computational Approaches to Code Switching*, 2014, pp. 21–31.

[26] S. N. Bhattu and V. Ravi, "Language Identification in Mixed Script Social Media Text," pp. 1–3, 2015.

[27] P. Pandey and S. Govilkar, "A Framework for Sentiment Analysis in Hindi using HSWN," vol. 119, no. 19, pp. 23–26, 2015.

[28] S. Sharma, P. Y. K. L. Srinivas, and R. C. Balabantaray, "Text normalization of code mix and sentiment analysis," *2015 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2015*, pp. 1468–1473, Aug. 2015.

[29] S. Sharma, P. Srinivas, and R. C. Balabantaray, "Sentiment analysis of code - mix script," in *Computing and Network Communications (CoCoNet), 2015 International Conference on*, 2015, no. 1967, pp. 530–534.

[30] R. Paper, P. P. Kulkarni, S. Patil, and G. Dhanokar, "Marathi And Sanskrit Word Identification By Using Genetic Algorithm," vol. 2, no. 12, pp. 4588–4598, 2015.