Domain-specific Sentiment Analysis Approaches for Code-mixed Social Network Data

Pravalika A, Vishvesh Oza, Meghana N P and Sowmya Kamath S
Department of Information Technology
National Institute of Technology Karnataka, Surathkal, Mangalore - 575025, INDIA
avvarupravalika, vishveshoza, meghananp@gmail.com; sowmyakamath@nitk.edu.in

***Abstract*—Sentiment Analysis is one of the prominent research fields in Natural Language Processing because of its widespread real-world applications. Customer preferences, options and experiences can be analyzed through social media, reviews, blogs and other online social networking site data. However, due to increasing informal usage of local languages in social media platforms, multi-lingual or code-mixed data is fast becoming a common occurrence. Mixed code is generated when users use more than a single language in social network comments. Such data presents a significant challenge for applications using sentiment analysis and is yet to be fully explored by researchers. Existing sentiment analysis methods applied to monolingual social data are not suitable for code-mixed data due to the inconsistency in the grammatical structure in these sentences. In this paper, a novel method focused on performing effective sentiment analysis of bilingual sentences written in Hindi and English is proposed, that takes into account linguistic code switching and the grammatical transitions between the two considered languages. Experimental evaluation using real-world, code-mixed datasets obtained from Facebook showed that the proposed approach achieved very good accuracy and was also efficient performance-wise.**

***Keywords*—Code-mixed data, Machine Learning, Sentiment analysis, grammatical transitions, lexical analysis.**

I. Introduction

Sentiment analysis is the use of computational linguistics and natural language processing techniques to extract subjective information from a source [1]. The source of any such pertinent information is primarily unstructured, that can be found in webpages, blogs, review websites or social networking sites. This type of data is continuously generated and is also of very high volume. Capturing the latest trends, the experiences of the customer and the overall opinion of users on a particular trending topic is hence a big data analytics problem that must deal with the huge volume, velocity and variety of social media data [2]. Businesses that want to market their product, want an unbiased opinion on market trends, consumer reviews etc, and sentiment analysis on user data provided by social media platforms like Twitter and Facebook is an effective way to achieve this [3].

Although English is still by far the most popular language in Social Media context, its dominance is already being challenged due to the rising popularity of regional dialects among native users. In 2011, Hong et al [4] applied an automatic language detection algorithm to over 62 million posts to identify the top 10 most popular languages on Twitter. Based on their findings, they reported that Hindi was the most popular language among Twitter users in India. As it is, India boasts several hundred existing languages, with high language diversity and dialect changes, which instigates frequent code mixing [5]. This actually compounds the problem, as, in any region of the world, people tend to use those languages which are most common to their region, in addition to English.

However, researchers have only started to consider the multilingual nature of social data and effective algorithms that can deal with this complexity are of paramount importance. Most existing works actually consider words other than English to be extraneous, which are removed during the preprocessing phases before actual sentiment analysis can be performed [6]. But, without analyzing these multilingual words in comments, the correct user sentiment cannot be perceived effectively. Traditional sentiment analysis algorithms that work well for monolingual data exhibit several shortcomings when applied to such multilingual, code-mixed social data [7]. Thus, there is huge scope for developing novel algorithms that leverage the concepts of grammatical transitions and code switching for code-mixed sentiment analysis.

In this paper, two different approaches for bilingual sentiment analysis for user data on Indian social media platforms, particularly Hindi-English mixed data are proposed. We considered English and Hindi, as these are the two languages most frequently used by social-media savvy people in India, resulting in code-mixing in social media interactions. In the first approach, a lexicon representing the *movie* domain is constructed that contains an extensive list of slang and abbreviated words in both English and Hindi, so as to ensure that most frequently used terms in social media like Facebook are captured. Using the sentiment of the words in the sentence as per the the lexicon, sentiment combination rules are inferred to judge the sentiment of the sentence. In the machine learning based sentiment analysis approach, the model is built by using mixed language training data obtained from Facebook, and its grammatical transitions and frequently occurring rules are learned. After extracting relevant features from the training set, the classifier is trained to detect polarity of a particular user comment.

The remainder of the paper is organized as follows. Section II briefly reviews existing work in the area of mono-lingual and multi-lingual sentiment analysis. Section III presents the proposed lexicon-based approach for Hindi-English code-mixed social data analysis. In Section IV, the machine learning approach for sentiment analysis of bilingual social data is discussed. In Section V, the results of the experimental validation and statistical analysis of the proposed methodology is discussed, followed by concluding remarks and directions for future research.

II. Related Work

Code mixing [5], though more commonly used in verbal communication, is increasingly found in social interactions between users of social networks like Twitter and Facebook. From a computational aspect, there have been limited studies on code mixing in social media. Analyzing multi-lingual social media content is very challenging because of the frequent usage of non-standard spellings and non-grammatical contractions between the used languages [8]. The linguistic complexity of such content is compounded by the presence of spelling variations, transliteration and non-adherence to formal grammar. This makes for a very interesting research problem that has motivated many researchers in the field of social data analysis.

Several works address the issue of performing sentiment analysis on monolingual data. A four step approach involving language identification, part-of-speech tagging, subjectivity detection and polarity detection was proposed by [9]. However, this method is only suitable for English language, and not for code-mixed data which exhibits linguistic code switching. In code-mixed data, a sentence can contain multiple languages in random order and hence only word by word language identification can be performed. This can lead to loss of context which is crucial in sentiment analysis [8].

Birbilli et al [10] presented another method for sentiment analysis of multiple languages based on document translation. However, translation itself causes serious problems including loss of conceptual equivalence as grammatical and syntactical structures of the source language cannot be compared. Narr et al [**?**] proposed a method of sentiment analysis for mixed languages where classifiers trained on different languages are used. The training data consisted of millions of posts in many languages like English, German, French and Portuguese. They used a Naive Bayes classifier on this training data to predict sentiment polarity. A semi-supervised heuristic labeling scheme was used to obtain a huge amount of training data in many languages and content-based features which can work well across languages.

Abbasi et al [11] presented a multilingual sentiment analysis approach that uses stylistic features of Arabic and English language for classification and opinion mining of English and Arabic web forms. Specific feature extraction components were integrated to account for linguistic characteristic of Arabic. Raghavi et al [12] proposed a question classification modeule for a full-fledged question answering system in code-mixed language. They used a Support Vector Machine (SVM) based approach for learning to identify code-mixing in English-Hindi questions submitted to the system.

Das et al [8] developed a mechanism for detecting language boundaries at the word level in chat message corpora in mixed English-Bengali and English-Hindi. They used a code-mixing index to evaluate the level of blending in the corpora and describe the performance of a system developed to separate multiple languages. Jamatia et al [13] developed a part of speech tagging system for Indian Language social data to deal with the issues of word-by-word analysis. This tool would have been helpful for identifying multilingual occurrences, however it is not available freely.

Sitaram et al [14] proposed a sentiment analysis method for code mixed data containing both English and Hindi, where Hindi was transliterated to English for the purpose of analysis. They used a recursive neural tensor network for classification based on word vector matrices in order to store the individual words and phrases. This was used to build a sentiment tree to assign sentiment for each word and phrase, while considering context. When compared to this approach, our proposed methodology used both lexicon and machine learning approaches, hence is more reliable when compared to purely machine learning based approaches. This is because, in sentiment analysis, it is important to consider the polarity of each and every word. Often, it is also not possible to transliterate all words from Hindi to English, which may lead to wrong polarity detection.

Bhargava et al[15], developed a method for sentiment analysis for mixed script indic sentences. It determines the overall sentiment of code mixed sentences for English and a combination of four other Indian languages (Tamil, Telugu, Hindi and Bengali). They solved the problem by identifying the language first and then applying a sentiment mining algorithm to get the overall sentiment. The algorithm finds the sentiment of tokens that are defined for the sentence and then calculates the overall sentiment based on positive and negative count of words in the sentence. We have developed a novel polarity detection algorithm which considers various other factors apart from the sentiment of the tokens to assign the polarity of the sentence. Our proposed method outperforms Bhargava et al's method and was able to achieve an accuracy of 86%.

When dealing with multilingual data, the grammatical changes or transitions have to be taken into consideration at the points where the languages change. For example, the sentence '*Yeh movie accha nahi hai*', is transcribed in English, but is a bilingual sentence. The sentiment expressed in the sentence is negative and there is a grammatical transition between English and Hindi which needs to be explicitly considered when correctly estimating the sentiment. Considering another example, '*Movie was so good ki main theater mein hi so gaya*' (which means that the movie was so boring that the watcher slept off in the theater), which is basically an ironical sentence that expresses a negative sentiment. However, if the code switching is not considered, this statement might be taken as a positive comment. Hence, the challenge is to develop an algorithm that can identify language shifts and predict the polarity of monolingual and bilingual sentences.

III. Proposed Methodology

The proposed system is a hybrid system that includes both lexicon-based and machine learning based approaches for calculating tweet sentiment from the semantic orientation of words present in the tweet. We describe the methodology defined in detail next.

A. Lexicon-based Approach

This approach uses dictionaries of words annotated with their semantic orientation (polarity) to classify text. This polarity classification task assigns a positive or negative label to a text that can capture the texts opinion towards its main subject matter. Figure 1 depicts the overall workflow of the lexicon based approach. The *Input* module takes a tweet as an input to determine its polarity. The *tokenizer* module tokenizes the input tweet and performs pre-processing. The *English Dictionary* and *Hindi Dictionary* are used, and consist of a list of words and corresponding polarity values compiled into a dictionary. The *Polarity Detection* module then applies the proposed algorithm on the tweet to predict the overall sentiment of the tweet.

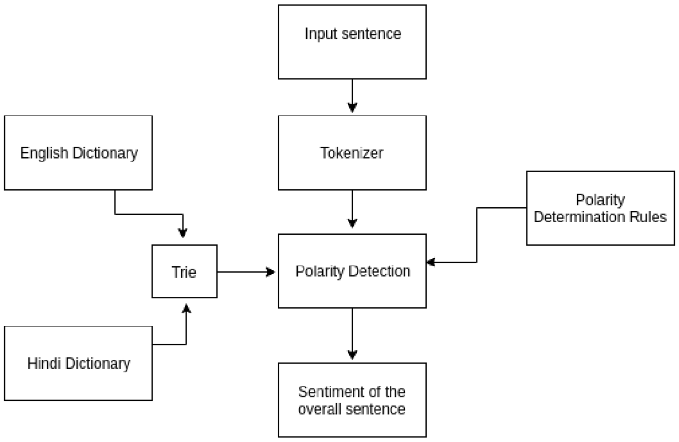

Fig. 1. Architecture of Lexicon Based Approach

1) Corpus Extraction And Data Pre-processing: The training data consists of 5000 posts on movies and it was collected using the Facebook API which were published between January to March 2016. Facebook was chosen as it is a more widely used social networking site than other social networking sites. Other sites might provide more security towards the personal data than Facebook, thus constraining the quality of data that can be extracted. The pages from which data was extracted were those which were very popular with 1.8 to 15.5 million "likes". Also the posts[1] having a long thread of comments (50+) were extracted because non-standard usage of language is more common in these comments. The corpus thus created had about 5000 posts and 18000 words. The data was semi-automatically cleaned and formatted.

In the case of opinions, all words used in the sentence don't have same significance. Some words are classified as noise as they don't affect the process of classifying the polarity of the post. During preprocessing, these noise words are removed by using regular expressions. Extra spaces, hyperlinks and escape sequences are also removed. Emoticons are ignored while forming the English-Hindi lexicon. Exaggeration is removed by replacing repeating letters in a word with thrice that letter, for example '*gooooooood*' is replaced with '*goood*'. It is replaced with thrice the letter and not twice because there are words where a letter can repeat twice so in order to prevent conversion of one word to another word with a totally different meaning, we convert or replace repeated letters with thrice that letter (in this case an extra '*o*').

2) English and Hindi Dictionary Creation: Figure 3 illustrates the process of dictionary creation. The dictionary can be created in different ways - manually, using existing dictionaries or semi-automatically, making use of resources like WordNet. In this paper, the dictionaries for lexicon-based approach were created manually. However, in light of the lack of stability/completeness of automatically generated dictionaries, we decided to create manual ones. These were produced by hand-tagging all words found in our development corpus, a 5000-text corpus of posts on a scale ranging from extremely negative to extremely positive, where 0 indicates a neutral word (excluded from our dictionaries). Positive and negative were decided on the basis of the words prior polarity, that is, its meaning in most contexts.

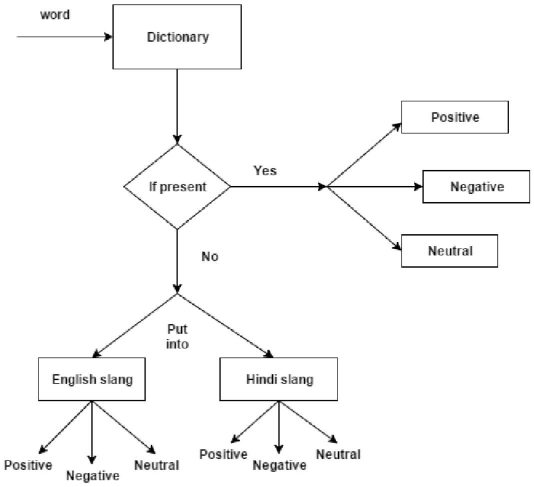

Fig. 2. Process of Dictionary Creation

A major concern in lexical based approaches is the fast extraction of words from the dictionary. This is achieved by implementing an efficient trie data structure [16] to store the dictionary. To tackle the issue on space and time complexity, we used a trie to store this data, instead of a conventional database. While determining the sentiment of the text, the words are extracted from dictionaries into a trie for efficient and fast search. A trie is an ordered tree data structure that is used to store a dynamic set where the keys are usually strings. Unlike a binary search tree, a node in the tree does not store the key associated with that node but its position in the tree defines the key with which it is associated. This helps in searching words very quickly and efficiently. This trie is used to store the English and Hindi words and their associated polarity is also stored for each word. The trie is initially populated with all the English words taken from Princeton dictionary (http://wordnet.princeton.edu/) while the Hindi word forms in their WX notation were populated from the dictionary provided by IIT Bombay (http://www.cfilt.iitb.ac.in/).

3) English And Hindi Slang dictionary: To tackle issues related to the usage of informal words, word shortening and spelling variations etc., in social network posts, a English and Hindi slang dictionary was also created and merged in the trie. A training corpus of 5000 posts was used to generate a lexicon that contains all the spell variations and slang words commonly used on social media. Language identification of the words in the post was necessary to assign their language tag before adding them to their respective slang dictionary. Algorithm 1 shows the process of using the trie for quick and easy retrieval of words from dictionaries.

Initially, the trie is loaded with words from English dic-

[1] The term '*post*' is used to refer to either a user's post or user's comment in this paper

tionary, Hindi dictionary, English senti-dictionary, Hindi senti-dictionary, English slang dictionary and Hindi slang dictionary before processing new words from the training data. Each new word is first searched in the trie, if it is not present then it is inserted into the appropriate dictionary based on the polarity and language manually. Addition of a new word to either the English or Hindi slang dictionary automatically adds the same word to the trie. This is to prevent addition of the repetitive slang words into the dictionary.

Algorithm 1 English & Hindi Slang Dictionary

```
procedure $Slang\_Dict\_Creation$
    load English Dictionary into a trie
    load Hindi Dictionary into the trie
    load Hindi Senti-Dictionary into the trie
    load Englsih Senti-Dictionary into the trie
    do
        check if token is present in trie
        if $token == present\&polarity! = assigned$ then
            assign polarity
        else
            if $token! = present\&langauge(token) == Hindi$ then
                add_to_Hindi_slang(token,polarity(token))
            else
                if $language(token) == English$ then
                    add_toEng_slang(token, polarity(token))
            add token to trie
    while $token\_array! = empty$
```

4) Polarity Generation Algorithm: Algorithm 2 depicts the process of generating the sentiment polarity using the phrases and sub-phrases in the source languages to produce sentences in mixed languages. The sentence extracts contains a particular sentiment. Ex: “*Lagta nahi*” and “*nahi hai*” are negative. To calculate the overall sentiment of the sentence, we combine the sentiment of these sentence extracts incrementally. We start with phrases and sub-phrases particular to a specific language, so that the grammatical structure specific to the language is conserved. Each of the groups or extracts (can be a single word or a group of words) are assigned a sentiment value (1 for negative, 2 for neutral and 3 for positive). When the groups are combined, the resultant sentiment depends on the sentiment of these groups. This can be imagined to be in the form of a tree, so the sentiment that remains at the root gives the overall polarity of the sentence.

Let S be an array that stores the strings of an input sentence. *cur_polarity* denotes the polarity of the sentence up to last considered token. P[] stores the polarity of the part of the sentence between two conjunctions or commas. Table I shows the sentiment calculation rules to be followed for assigning polarity, when applying the procedure.

Algorithm 2 Polarity Detection Algorithm

```
procedure $SentimentDetection$
    input the sentence
    tokenize the sentence and store it in array S
    Initialize cur_polarity to 0
    do
        check polarity of the token from trie
        if $token = conjunction$ then
            add cur_polarity in array P
        else
            if $token$¬neutral & cur_polarity = 0 then
                cur_polarity ←token polarity
            else
                if $token$¬neutral then
                    cur_polarity ←token polarity * cur_polarity
    while $S \neq empty$
    do
        $polarity \leftarrow polarity + 1$
        $i \leftarrow i + 1$
    while $P \neq empty$
    Sentiment of Sentence $\leftarrow polarity$
```

TABLE I. SENTIMENT CALCULATION RULES

Polarity		Combined Polarity
Token 1	**Token 2**	
Positive	Positive	Positive
Positive	Negative	Negative
Negative	Negative	Negative

5) Sentence Polarity Calculation: On applying the Polarity prediction algorithm to the testing data, we note that the polarity of the sentence is the sentiment obtained on adding the polarity of all the phrases or sub-phrases. If it is 1 then the sentence is negative, if 2 then neutral, if 3 then positive. For example, considering a post *“Wo movie jabardast he”*, on applying the Polarity Detection Algorithm, we get the scores assigned to each of the constituent words as follows: {*Wo(0) + movie(0) + jabardast(1) + he(0)*} Here, we get an array which stores 1 and hence the sentence is positive. Considering an example with a conjunction *“Wo movie creative and achi he”*, on processing this we get an expression {*wo(0) + movie(0) + creative(1) + achi(1)+ he(0)*}. Here, the array stores 2 elements consisting of 1 each. The overall polarity for this post will be 2 and hence this is again positive.

B. Machine Learning Based Approach

Most statistical text classification approaches employ machine learning classifiers, trained on a particular dataset using features such as unigrams or bigrams, and with or without part-of-speech labels, although the most successful features seem to be basic unigrams. This approach involves data collection, data preprocessing, feature creation and lastly, application of machine learning techniques to train the classifier. We attempt this approach too for a comparative evaluation against the lexicon based approach.

1) Data Collection: The 5000 posts which were used for the lexical based approach were also used for training the classifiers. Extracting features directly from Facebook data is difficult due to misspellings and slang words. Hence, a preprocessing step is performed before feature extraction. Preprocessing steps include removing URL, avoiding misspellings and slang words. Misspellings are avoided by replacing repeated characters with 2 occurrences. Slang words add a lot of emotion to a post, so they shouldn't be removed. Therefore, a slang word dictionary is maintained to replace slang words present in the post with their associated meanings. Domain information which is movies contributes much to the formation of slang word dictionary.

2) Feature Identification: After extracting and removing post-specific features, the unigram approach is applied for tokenizing the posts to represent them as a collection of words. In unigrams, a post is represented by individual words. Lists containing negative words, positive words and negation words are maintained. Counts of positive and negative words in the posts are used as two different features in the feature vector, where presence of negation also contributes to the sentiment, hence negation is considered as a feature.

All words cannot be treated equally in the presence of multiple positive and negative words. Hence, a special word is selected from all the posts. In the case of posts having only positive words or only negative words, a search is done to identify a word having relevant part of speech. A relevant part of speech can be either an adjective, adverb or verb. Such a relevant part of speech is defined based on their relevance in determining sentiment. This is because, words that are adjective, adverb or verb shows more emotion than others. If a relevant part of speech can be determined for a word, then that is taken as special word. Otherwise, a word is selected randomly from the available words as special keyword. If both positive and negative words are present in a tweet, we select any word having relevant part of speech. If relevant part of speech is present for both positive and negative words, none of them is chosen. Such special word features are given a weight of '1' if it is positive and '-1' if it is negative and '0' in its absence. Part of speech feature is given a value of '1' if it is relevant and '0' otherwise.

3) Sentiment Classification: After creating a feature vector, Naive Bayes(NB), Support Vector Machine(SVM), Decision Tree(DT), Random Tree(RT), Multilayer Perceptron were applied to the data for performing sentiment classification. The WEKA tool is used to perform these classification on the training data, the experimental results of which are presented in detail in Section IV.

IV. Experimental Results and Analysis

The dataset used for the evaluation of the proposed approach consisted of 5000 user comments/post regarding movie reviews, extracted from Facebook API. Preprocessing techniques are described in Section III are applied and the data is cleaned. At present, we make an assumption that these posts do not have ironical or sarcastic comments. Our current approach can handle only non-sarcastic sentences, but we intend to extend our model in future to also address the presence of irony and sarcasm in user comments. Any emoticons used by the users in the comments are removed as noise as our approach automatically captures sentiment by analyzing the polarity of words used in the comments. Thus, the overall sentiment of the sentence is not affected or influenced by emoticons. The data is domain dependent so that all the words including informal words and commonly misspelled words are captured from the training data.

Now, for the lexicon based, the English and Hindi slang dictionaries can be created using these 5000 posts. Finally, the trie is loaded with the English, Hindi, English-senti, Hindi-senti, English-slang and Hindi-slang words from their respective dictionaries. It was observed that, the usage of the trie reduced the search time drastically, as search operations in tries are $O(1)$, thus making the algorithm computationally efficient. For the machine learning based approach, the same dataset is divided into training and testing data as per the 80:20 ratio. The features are extracted as described in Section III.B, and machine learning algorithms are applied to the dataset.

Table II depicts the results obtained on applying lexical approach on a set of 200 sentences. Around 95 out of 103 positive sentences were detected correctly, 52 out of 61 negative sentences were detected correctly and 29 out of 39 neutral sentences were detected correctly, resulting in an the overall accuracy of 88%. Table III presents the performance of various classifiers on the code-mixed data set. The data consists of sentences which are classified as negative , positive and neutral. It can be observed that SVM classifier has performed the best as SVM maps the data points to a higher dimension where the data becomes linearly separable. This is followed by Random Forest. Figure 3 shows the comparative performance of these classifiers, w.r.t precision and recall.

TABLE II. Analysis of lexicon approach for 200 sentences

Sentiment Class	Correct	Incorrect	Total
Positive	95	8	103
Negative	52	6	61
Neutral	29	10	39
Total	176	14	200

TABLE III. Recall and precision

Classifier used	Precision	Recall
Naive Bayes	0.725	0.735
Multilayer Perceptron	0.695	0.702
Random Tree	0.752	0.755
SVM	0.718	0.77
Decision Tree	0.701	0.723

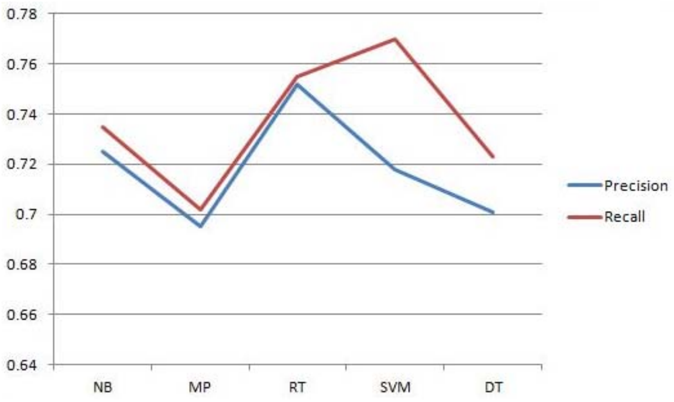

Fig. 3. Comparison of different machine learning algorithms

To compare the performance of lexicon based and machine learning based approaches, a testing data of 50 new posts was used. These sentence extracts (i.e. the phrases or subphrases belonging to either of the source languages) usually contains a particular sentiment. For example, "lagta nahi" is negative, "we Formula ready" is positive, "defensive captain" is negative, and "nahi hai" is negative. To calculate the overall sentiment of a sentence we combine the sentiment of these sentence extracts incrementally. We start by grouping phrases and sub-phrases particular to one language, to take care of the grammar specific to each language. Then, we group these language specific groups to take care of grammatical structure transition. The final group covers the whole sentence, and each group is assigned a sentiment value.

TABLE IV. ANALYSIS OF LEXICON APPROACH FOR 50 SENTENCES

Type	Correct	Incorrect	Total
Positive	13	5	18
Negative	14	1	16
Neutral	16	1	17
Total	43	7	50

Table 4 shows the analysis of machine learning approach. Out of 18 positive sentences, 14 were detected as positive and 4 were detected as negative. Hence, 14 out of 18 were correctly guessed and 4 were incorrectly detected. Also, out of 16 negative sentences, 12 were correctly detected as negative and 4 were incorrectly detected as not negative. Out of 16 neutral sentences, 10 sentences were detected to be neutral whereas 6 were detected as not neutral, resulting in an accuracy of 72%. From table IV and V, it can be observed that lexical approach achieved an accuracy of 86%, which is significantly higher than the machine learning approach.

TABLE V. ANALYSIS OF MACHINE LEARNING APPROACH FOR 50 SENTENCES

Sentiment class	Correct	Incorrect	Total
Positive	14	4	18
Negative	12	4	16
Neutral	10	6	16
Total	36	14	50

V. CONCLUSION AND FUTURE WORK

In this paper, two approaches for sentiment analysis of Hindi and English code-mixed data were presented. In the first approach, a lexicon representing the *movie* domain is constructed that contains an extensive list of slang and abbreviated words in both English and Hindi along with the standard English and Hindi words. This was done to ensure that most frequently used terms in social media are captured. Using the sentiment of the words in the sentence as per the constructed lexicon, sentiment combination rules are inferred to judge the sentiment of the sentence. In the machine learning based sentiment analysis approach, the model was built by using mixed language training data obtained from Facebook to learn its grammatical transitions and frequently occurring rules. After extracting relevant features from the training set, the classifier was trained to detect polarity of a particular user comment. Experimental evaluation showed that the proposed approaches performed well for real-world code-mixed social data. The lexicon-based approach achieved the best accuracy of 86%, while the accuracy for the machine learning approach was about 72%. As part of future work, we intend to extend the lexicon-based approach to make it domain independent, and also improve the composition of the training dataset to achieve better accuracy with the machine learning approach. We also plan to extend the scope of our approach by addressing multilingual code-mixed social data, without restricting to just Hindi and English.

REFERENCES

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[2] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, vol. 50, pp. 723–762, 2014.

[3] M. Rambocas, J. Gama *et al.*, "Marketing research: The role of sentiment analysis," Universidade do Porto, Faculdade de Economia do Porto, Tech. Rep., 2013.

[4] L. Hong, G. Convertino, and E. H. Chi, "Language matters in twitter: A large scale study." in *ICWSM*, 2011.

[5] U. Barman, A. Das, J. Wagner, and J. Foster, "Code mixing: A challenge for language identification in the language of social media," *EMNLP 2014*, vol. 13, 2014.

[6] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, 2005, pp. 347–354.

[7] E. Tromp, *Multilingual sentiment analysis on social media*. Lap Lambert Academic Publ, 2012.

[8] A. Das and B. Gambäck, "Identifying languages at the word level in code-mixed indian social media text," 2014.

[9] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[10] M. Birbili, "Translating from one language to another," *Social research update*, vol. 31, no. 1, pp. 1–7, 2000.

[11] H. C. Abbasi, Ahmed and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums," *ACM Transactions on Information Systems (TOIS)*, vol. 26, 2008.

[12] K. C. Raghavi, M. K. Chinnakotla, and M. Shrivastava, "Answer ka type kya he?: Learning to classify questions in code-mixed language," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 853–858.

[13] A. Jamatia and A. Das, "Part-of-speech tagging system for indian social media text on twitter," in *Proceedings Workshop on Language Technologies For Indian Social Media (SOCIAL-INDIA)*, 2014, pp. 21–28.

[14] D. Sitaram, S. Murthy, D. Ray, D. Sharma, and K. Dhar, "Sentiment analysis of mixed language employing hindi-english code switching," in *International Conference on Machine Learning and Cybernetics*, 2015.

[15] R. Bhargava, Y. Sharma, and S. Sharma, "Sentiment analysis for mixed script indic sentences," in *Conference on Advances in Computing, Communications and Informatics*, 2016.

[16] J.-I. Aoe, K. Morimoto, and T. Sato, "An efficient implementation of trie structures," *Software: Practice and Experience*, vol. 22, no. 9, pp. 695–721, 1992.