Word Level Language Identification in Code-Mixed Data using Word Embedding Methods for Indian Languages

Inumella Chaitanya, Indeevar Madapakula, Subham Kumar Gupta, Thara S
Dept of Computer Science and Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
Email: thara@am.amrita.edu,
ichaitanya234@gmail.com,subhamg145@gmail.com, iindeevar129@gmail.com

***Abstract*—In recent years, social media networking has grown to be a marvel of technology in our way of life. Facebook operates the world's leading web-based social networking system with over 2.19 billion clients(as of the first quarter of 2018). As its popularity increased, more individuals from all age demographics, have been accessing this growing phenomenon. Resultant usage of code-mixed data has become an all too common practice in the context of social media. The aim of our project was to identify different languages in the processing of code- mixed data. A comparison of different word embedding methods like Continuous Bag of Words (CBOW) and Skip-Gram models was used to generate feature vectors. These vectors are given as input to the machine learning algorithms like Support Vector Machine, Logistic Regression, K-Nearest Neighbors, Gauss Naive Bayes, Adaboost, and Random Forest which yielded in good cross-validation scores. The paper also reveals that Precision, Recall, F-Score, Micro and Macro averaging were used as evaluation measures.**

Keywords—Code mixed, Word2Vec, Continuous Bag of Words, Skip-Gram, feature vectors,Support Vector Machine , Logistic Regression, k-Nearest Neighbors, Gauss Naive Bayes, Adaboost and Random Forest.

I. Introduction

Code mixing can be simply interpreted as mixing of two or more languages in the conduct of communications. It is quite common for a speaker who knows two or more languages, to intermix vocabulary from different languages. Both the terms code-mixing and code-switching have been used interchangeably, is to highlight mixing of words from multiple languages in the communication process.

Social networking websites like Facebook, Twitter, WhatsApp, etc. are being widely used by people to stay in touch on a daily basis. India has 23 official state languages, besides numerous unofficial languages. While conversing, an Indian, who speaks English and Hindi, shows the general impact of Hindi with English words followed by the other regional native languages. Code mixing can be defined as usage of two or more languages in the exchange of information between 2 or more individuals. It is a common observation that a speaker, who is familiar with more than one language, interjects few words, phrases or sentences from another language. Here is an instance of code-mixed Hindi and English, "Mein ne kal movie dekha and it was very good". In plain English, this sentence can be translated as 'I saw a movie yesterday, and it was very good'

In modern day communications, where social media platforms like Facebook, Twitter, WhatsApp messages, have made available scores of communicative options usage of code-mixed languages, has set in many new challenges in identification of languages at the word-level . As most used language identifiers or detectors are hard to find languages in social media text because of writing style, rather than a general assumption is language identification is already solved problem. Das, et al in [2], observed that multi-lingual speakers, while conversing in a language, say English, do not use a single code to speak or write in their native language; they often use phonetic typing or randomly merged English and native words (through code-mixing). In general, trying to make identification of a language in the domain of social media communications can present convoluted challenges.

The remainder of the paper is organized as follows: Section II describes about the literature survey that has been done in the Part Of Speech (POS) tagging in code mixed data. Section III describes about the proposed method, data-set description, preprocessing stages, word embedding models and the machine learning algorithms used for comparison. Section IV discusses about experimental results, evaluation measures and the cross validation scores obtained. Section V gives the conclusion of the proposed work and gives path to future work.

II. Literature Survey

Das and his team [2] developed a system that was improved to automatically identify boundaries of language in code-mixed social media data. It describes a simple dictionary based approach as its baseline system. The system includes some feature extraction methods such as N-gram pruning + dictionary, word context + Dictionary + N-Gram Weight were used for SVM classification. The performance is measured on a 10 fold cross validation.

In paper [3] Paul et.al, discusses a novel method of incremental POS tagging to code-mixed text. Their paper uses dynamic model switching, and has got an indicator function which emits term-by-term language identification tags. It has even got a controller which screens the output and chooses the most suitable tagging model to use for a given term. It uses a corpus of Spanish/English conversation data for experimental purposes. The baseline system achieved an overall accuracy of 77.27%.

In [4] Barman et.al, talks about language identification problem in the contexts of code-mixed social media. Their paper provides a new dataset such as facebook posts wherein the comments between English,Bengali and Hindi are considered. Numerous methodologies were employed which dictionary-based approach in an unsupervised algorithm, word-level identification which is supervised by using and without using contextual information and also sequential labelling approach such as condition random Fields. Few machine learning toolkits used include WEKA [7], MALLET [8], LibLinear [9], and NLP tools [10].

The paper [5] Sarkar et.al, described approaches to the Part-of-Speech (POS) tagging techniques that was exploited in the investigation reported herein. Their paper proposed a POS tagger based on Hidden Markov Model (HMM). It uses information from dictionary based approaches and some word level features to further improve the observation probabilities for prediction. It has been trained and tested on ICON 2015 shared dataset of Bengali-English, Hindi-English, Tamil-English. The developed POS system has been evaluated with some traditional evaluation measures like Precision, Recall and F1 measure.

Muysken [6], explained that POS tagging is concerned with the task of allocating grammatical categories (noun, adjective, verb etc.) to words in a natural language sentence. POS tagging can be effectively utilized in various Natural Language Processing (NLP) applications. The main focus in using NLP methods for checking no standardized texts, such as mainly social media texts, which is growing rapidly, because the automatic analysis of social media texts is one of the essential requirements for the task of sentiment analysis. Their paper capitalizes on the POS tagging system for code-mixed text in Indian languages, collected from social media.

Dong et.al, in [11] dealt with automatic identification of languages at the word level in code-mixed dataset. The data for their paper comes from Turkish-Dutch speakers. The experimental setup includes Language Models(LM),Dictionary + Language Models(DICT+LM). The classifiers used in their paper includes Logistic Regression (LR) and Condition Random Fields(CRF) and it also gives importance to the context for enhancing the performance.

III. Proposed Methodology

The Proposed Method has been represented as a flow chart as shown in the Figure 1. The stages are explained as follows:

A. Pre-Processing

Pre-processing of data is a crucial step in any Natural Language Processing (NLP) tasks. The raw data available is always in unstructured format so with the help of Natural Language Tool Kit (NLTK) package in Python language we can convert it into structured and intelligible form. Few major pre-processing stages include removing of URLs, emoticons, hypertext links, hashtags, punctuation's etc. By using regular expression in NLTK package we can remove all these unwanted expressions.

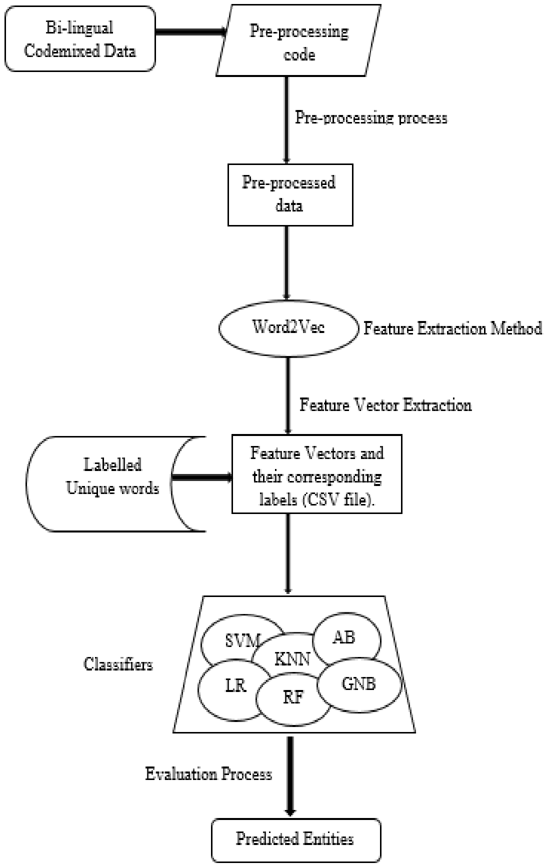

Fig. 1. Proposed Diagram

Algorithm 1 Pseudo code for the proposed method

```
Input  : Code-Mixeddata ← Data[]
Output: Predicted entities(Language) for each word.
//Removes unwanted data like Emoji's, URL's, hashtags, etc., in the data.
//Feature_vector_extraction method(data): This method consists of feature vector extraction methods(Word2Vec) to which the pre-processed data is passed as a parameter.
//Unique_words(data): Returns a list of unique words from data.
//Label(i): Returns the prescribed label of a word i.
//trained_data: This variable holds the labels of each unique word.
//make_CSV(a, b): Creates a CSV file with a as the first column and b as the second column.
//Classifier(word, file): This method consists of 6 classifiers(SVM, Random Forest, KNN, Logstic Regression, Gaussian Naive Bayes and Adaboost) to which a test word and the csv file are passed as parameters.
for i = 0  to Data.Length()  do
    pre_ProcessedData ← $Pre\_processing(Data[i])$
end for
$fVec \leftarrow FeatureExtractionAlgorithm$
$(pre\_ProcessedData)$
totalWords ← $Unique\_words(preProcessedData)$
for i = 0  to totalWords.Length()  do
    trained_data ← $trained\_data + label(i)$
end for
Test.csv ← $make\_CSV(fVec, trained\_data)$
Predicted_Language ← $Classifier(testWord, Test.csv)$
```

B. Word Embedding Models

The next step is to send the pre-processed data as input to the required word embedding model. Word embedding is nothing but word vectors or vectors of weights. Suppose we have some d-dimensions then each word can be represented in these d-dimensions. Each word will have different weights in different dimensions. Word Embeddings are useful for representing the meaning of words for different applications. In our research, the word embedding model used is Word2Vec which is a combination of two models, namely Continuous Bag-of-Words (CBOW) and Skip-Gram. In CBOW method we are using the neighbouring words and trying to predict the centre/target word. One disadvantage of CBOW method is that it is having high computational cost if corpus is really large (huge) because we need to compute the summation over all words in each iteration whereas in small corpus the result is not so good when compared with large dataset. CBOW is illustrated as shown in Figure 2.

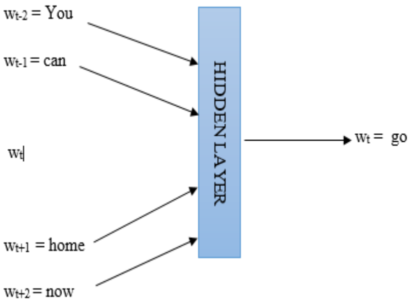

Fig. 2. Representation of CBOW technique

For simplicity, let us consider a target word and then we are trying to pick a window size of 5 for the context words . From the example shown: ”You”, ”can”, ”home” and ”now” are the neighbouring words and these words it should be able to predict ”go” which is the target word. The training objective is we are trying to maximize the conditional probability of observing the actual output word given the input context words, with regard to the weights. Mathematically speaking the input to CBOW model will be (w_tw t - 2), (w_tw t - 1), (w_tw t + 1), (w_tw t + 2) the preceding and following words of the current word and the output will be w_tw t.

The skip-gram model is the opposite of the CBOW model. It is constructed with the focus word as the single input vector, and the target context words are now at the output layer as shown below in the Figure 3.
Mathematically speaking Skip-gram model requires only a single input i.e.w_tw t and the output will be (w_tw t - 2), (w_tw t - 1), (w_tw t + 1), (w_tw t + 2) the preceding and following words of the current word

C. Algorithms used for Classification

The next step is classification and the feature vectors obtained from both CBOW and Skip-gram model are given as input to six different classifiers and a comparison study is performed. The different classifiers used for comparative study include:

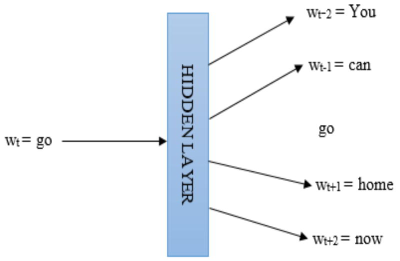

Fig. 3. Representation of Skip-Gram technique

1) Support Vector Machine (SVM)
2) Random Forest (RF)
3) Logistic Regression (LR)
4) Gaussian Naive Bayes (GNB)
5) k-Nearest Neighbour (KNN)
6) Adaboost

Support Vector Machine is a Supervised Classification algorithm. This means that SVM needs training data [15]. Given a set of train data and a set of corresponding class labels SVM can used to create a model. Using this model, it predicts the class of a new testing sample. The objective behind SVM is to find the optimal hyper plane which separates the training data. In two dimensions the hyper plane can be visualized as a line. An optimal hyper plane is the one which maximizes the margin of training data [13]. SVM uses the concept of hyper plane to classify the data.

The second type of classifier is Random Forest [14]. It actually resolves the limitation issues faced by decision tree. It mainly aims to make the trees de-correlated and trim the trees by setting some criteria for stopping the node splits. The third model of classification is Logistic regression. It classifies the data based on the probabilities, where you are required to set the probability cut-off for classifying the data. The fourth type of classifier is Gaussian Naive Bayes algorithm which is a probabilistic classifier. It is based on probability models that incorporate strong assumptions. The fifth type of classifier is K-NN, it is a non-parametric method used for classification and regression. In K-NN classification [16], the output is a class membership. It stores the available cases and classifies new cases based on a measure of similarity. The sixth model is AdaBoost [16], it is best used to boost the performance of decision trees on binary classification problems. The output of the remaining learning algorithms is mixed to give a weighted sum that shows the final output of the classifier which is boosted. Adaboost is adaptive in the sense that weak learners are twisted in favour of those instances which are misclassified by previous classifiers. AdaBoost is very much sensitive to noisy data and outliers. After applying classification with different classifiers, their outputs are analysed and compared.

IV. Experimental Results

A. Data set description

The ICON 2016 dataset [12] is used for our project. It consists training data of social media sites like Facebook(1K),Twitter(1K), WhatsApp (1K) for Hindi-English language pair. For example

$$”Yaar, \ \ tu \ \ good \ \ hai”$$

. Words in this text, are: Yaar, tu, god, hai. Despite the fact that code-mixing is a natural practice for multilingual, we are unaware of the distribution profile of code-mixing in any social-media corpus. The dataset description is show in the TABLE I.

TABLE I. Data set description

Method	No of words	Languages	No of entities
Word2Vec	7210	HIN, ENG	2

B. Experiment

The first stage of our work starts with pre-processing of data, i.e removing the unwanted data like emojis, hypertext links, etc. This process is achieved by using NLTK packages in python. This pre-processed data is given as an input to Word2Vec algorithm. The feature vectors for a dimension of 100 with a learning rate of 0.000001 were generated for CBOW and Skip-Gram algorithms separately and also we have taken a window size of 5 for both of them. There were totally 6291 unique words generated by this algorithm. The number of unique words generated for this algorithm were 6970 as Stop words were used in this algorithm. Labelling of all the unique words was done manually. These feature vectors were combined with their corresponding labels into a CSV file. This file was given as input to all the six classifiers. The cross validation scores of all the classifiers were taken as the accuracy values.

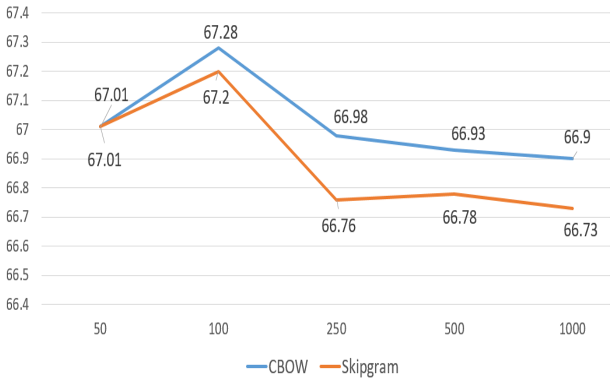

Fig. 4. Accuracy of CBOW and Skip-Gram at different Dimensions

C. Evaluation Measures

They are two types of classes i.e. Actual and Predicted. In Actual class we have truth value for the ground values of the class label whereas in the Predicted class we have the predicted values.

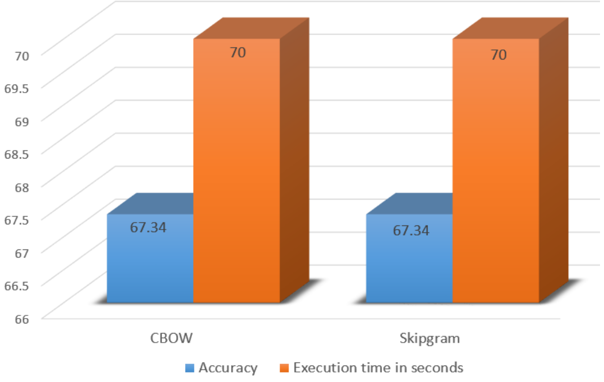

Fig. 5. Accuracy and Execution time (in seconds) for CBOW and Skip-Gram

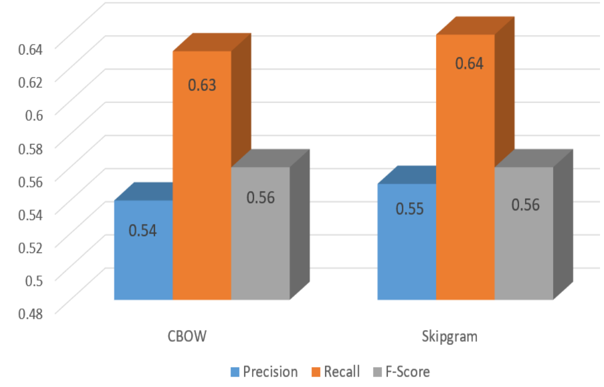

Fig. 6. Precision, Recall and F-Score for CBOW and Skip-Gram

The following are the 3 evaluation measures:
Precision:

$$Precision = \frac{(TP)}{(FP + TP)} \tag{1}$$

where FP is False Positive and TP is True Positive.
Recall:

$$Recall = \frac{(TP)}{(TP + FN)} \tag{2}$$

where TP is True Positive and FN is False Negative.
F-score:

$$F - score = 2 \ (\frac{Recall * Precision}{Recall + Precision}) \tag{3}$$

D. Results

Figure 4 shows the accuracy of Word2Vec models i.e CBOW and Skip-Gram models at different dimensions ranging from 50 to 1000. Figure 5 shows a plot of accuracy versus execution time. From the Accuracy bar chart, we can see that Skip-Gram and CBOW shows a similar accuracy level. The variation is very small because we are using a small data set of 7210 words. Table IV gives the tabular representation of various classifiers where SVM gives the highest accuracy.

TABLE II. Predicted & Actual classes in the columns & rows

	Yes	No
Yes	TP	FN
No	FP	TN

TABLE III. Cross validation scores of various classifiers

Classifiers	**Using CBOW (in %)**	**Using Skip-gram (in %)**
SVM	67.33	67.34
Random Forest	58.67	59.65
Logistic Regression	58.67	59.65
GNB	67.19	66.98
KNN	59.87	60.45
Adaboost	67.01	67.28

V. Conclusion

Through this paper, we want to prove that Code-Mixing is not a grammar-less phenomenon with the use of Word2Vec [17] which is a word embedding method. Our prime objective of this study was to improve accuracy of the word level identification of language in a code-mixed data by using Word2Vec approach. With the use of traditional algorithms we made a comparison study of giving CBOW and Skip-Gram as input to these algorithms at different dimensions. Precison, recall and F-measure were also calculated. For future work we could extend it with more Indian Languages and also its accuracy can be enhanced with the help of neural network.

References

[1] Barman, Utsab, Amitava Das, Joachim Wagner, and Jennifer 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*, pp. 13-23. 2014.

[2] Das, Amitava, and Bjrn Gambck. ”Identifying languages at the word level in code-mixed indian social media text.” (2014).

[3] Rodrigues, Paul, and Sandra Kbler. ”Part of Speech Tagging Bilingual Speech Transcripts with Intrasentential Model Switching.” In *AAAI Spring Symposium: Analyzing Microtext*. 2013.

[4] Barman, Utsab, Amitava Das, Joachim Wagner, and Jennifer 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*, pp. 13-23. 2014.

[5] Sarkar, Kamal. ”Part-of-speech tagging for code-mixed indian social media text at ICON 2015.”*arXiv preprint arXiv:1601.01195* (2016).

[6] Muysken, Pieter Cornelis. ”Code-switching and grammatical theory.” (1995).

[7] Weka 3: Data Mining Software in Java. Available: http://www.cs.waikato.ac.nz/ml/weka/.

[8] McCallum, Andrew Kachites. ”Mallet: A machine learning for language toolkit.” (2002).

[9] Thara, S. ”Code-Mixing: A Brief Survey” In *Advances in Computing, Communications and Informatics (ICACCI), 2018 International Conference on*, pp. IEEE, 2018.

[10] Owoputi, Olutobi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. ”Improved part-of-speech tagging for online conversational text with word clusters.” In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 380-390. 2013.

[11] Nguyen, Dong, and A. Seza Doruz. ”Word level language identification in online multilingual communication.” In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 857-862. 2013.

[12] Jamatia, Anupam, and Amitava Das. ”Task report: Tool contest on POS tagging for codemixed Indian social media (Facebook, Twitter, and Whatsapp) Text@ icon 2016.” *the proceeding of ICON 2016* (2016).

[13] Thara, S. ”A Comparison Study of Word Embedding for Detecting Named Entities of Code-Mixed Data in Indian Language” In *Advances in Computing, Communications and Informatics (ICACCI), 2018 International Conference on*, pp. IEEE, 2018.

[14] Thara, S., and S. Sidharth. ”Aspect based sentiment classication: Svd features.” In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pp. 2370-2374. IEEE, 2017.

[15] Thara S, Athul krishna,”Aspect Sentiment Identification using random Fourier features”, International Journal of Intelligent Systems and Applications(IJISA), 2018.

[16] Kumar, S. Sachin, K. Manjusha, and K. P. Soman. ”Novel SVD based character recognition approach for Malayalam language script.” In *Recent Advances in Intelligent Informatics*, pp. 435-442. Springer, Cham, 2014.

[17] Kumar, S. S., M. Anand Kumar, and K. P. Soman. ”Experimental analysis of Malayalam POS tagger using Epic framework in Scala.” *ARPN J. Eng. Appl. Sci 11* (2016).

[18] https://code.google.com/archive/p/word2vec/