Language Identification in Mixed Script

Nagesh Bhattu Sristy
CoE on Analytics, IDRBT
Hyderabad, India
nageshbs@idrbt.ac.in

N. Satya Krishna
CoE on Analytics, IDRBT
Hyderabad, India
satya.krishna.nunna@gmail.com

B. Shiva Krishna
CoE on Analytics, IDRBT
Hyderabad, India
shivakrishnaboora@gmail.com

Vadlamani Ravi
CoE on Analytics, IDRBT
Hyderabad, India
vravi@idrbt.ac.in

ABSTRACT

The text exchanged in social media conversations is often noisy with a mixture of stylistic and misspelt variations of original words. Any standard NLP techniques applied on such data such as POS tagging, Named entity recognition suffer because of noisy nature of the input. Usage of mixed script text is also prevalent in social media users. The current work addresses the identification of language at word level in mixed script scenarios, where all the text is written in roman script but the words being used by the users are transliterations of original words in native language into *english*. The core part of the problem is identifying the language, looking at small fragments of text among a set of languages. We propose a two stage approach for word-level language identification. In the first stage a mixing language combination is identified by using character n-grams of the sentence. Second stage consists of using the previous mixing combination class to make the word level language identification. We apply Conditional Random Fields(CRF) further in second stage to improve the performance of the word level language identification. Such simplification is essential, otherwise the number of states of the model will be huge and resultant model predictions are very noisy. Our methods improve the F-score of word level language identification by over 10% compared to the base-line.

CCS CONCEPTS

• **Computing methodologies → Discourse, dialogue and pragmatics**;

KEYWORDS

Language Identification, CRF

ACM Reference Format:
Nagesh Bhattu Sristy, N. Satya Krishna, B. Shiva Krishna, and Vadlamani Ravi. 2017. Language Identification in Mixed Script . In *FIRE'17: Forum for Information Retrieval Evaluation, December 8–10, 2017, Bangalore, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3158354.3158357

1 INTRODUCTION

Recently social media based tools have revolutionized the way people interact with one another, facilitating users to readily share data with others. Going further these tools opened a set of new opportunities for all segments of businesses across the globe, to reach out prospective customers, to advertise products through campaigns etc. There are many challenges in building automated tools for processing the text primarily exchanged in these tools, because of free form of language being used by the users. Many techniques are proposed in the recent past for addressing the noisy form of text exchanged in the social media text. The number of tools for rich language like *english* are plenty compared to the rest. The tools available other languages too, can be tweaked to address the noisy version of social text. But the difficulty lies in addressing the large state-space of models as the number of correct variations for a textual entity borrowed from a different languages would be too many.

In this work we address the problem of identifying the language in the context of code-mixing. Code-mixed communication is prevalent among social media users, as a means for communicating with others in short spans of text and also for effectively expressing one's opinion. Majority of such code-mixing is written in the same romanized script as switching between different keyboard interfaces is difficult. The analysis of code-mixing between *hindi* and *english* in Facebook posts is analyzed in [28]. They have analyzed difficulty of distinguishing the words written in roman script. As there are some words which are actually in *hindi* but the same spelling is valid and makes sense in *english* also. So those have to be handled differently.

Code-mixing or code-switching is studied from linguistic perspective in many studies [25],[23]. These studies characterize the code-mixing as broadly two types namely inter-sentential and intra-sentential code-switching. Muysken [23] reserves code-mixing for intra-sentential. But we use these terms interchangeably as they are two different cases of the same problem from a computational perspective.

Machine learning algorithms are proposed in the recent past to address the tagging of words with the language identifier. Language identification tools such as *langid.py* [19] address the problem and solve it at sentence level using different classification algorithms.

Most of the other approaches studied in the literature address the problem of language identification where the two mixing languages follow two different scripts. McNamee [22] show how simple n-gram based methods can be used to solve such problem upto 100% accuracy. Language identification is closely related to the process of language analysis in [30] and it has showed applications of machine learning methods using n-gram based features. This study [30] is done at sentence level language identification. In the current study we address the problem of language identification at word level.

Word level language identification is comparatively harder in code-mixed scenario, due to the presence of numerous ways of spelling a single word in non-native language. For example the word 'pehala' in *hindi* (meaning 'first' in *english*) can be written in 'pehela', 'pahala','pahalaa', 'pahila', etc.. A problem that closely addresses the difficulty posed by numerous variations of single word is called spelling correction. Spelling correction systems use specialized models [17] to address this through special hidden states to allow for the numerous variations.

The availability of data with this nature also poses significant challenge for making progress in this setting. Semi-supervised [6] and weakly supervised [10] approaches make significant impact in this scenario to make use of large amount of unlabeled data for training better classifier. In this work we use a two stage approach for the word-level language identification problem. First stage consists of identifying the mixing language combination and second stage consists of application of language combination specific classifier to distinguish between words of the two mixing languages. We further applied a sequence labeling algorithm such as CRF to improve the word level language identification of successive words.

2 RELATED WORK

Eskander et al. [9] have presented a system that identifies the word as *arabic* language identification. The feature set used in this system include different combinations of n-grams, word itself, word length and some probabilistic features. Carpuat [5] identified mixed language segments in *french* and *english* in Canadian Hansard parallel corpus. Word level tagging is done using CRF. By using CRF Jain et al. [15] identified code-switching in four language pairs namely *nepali-english*, *spanish-english*, *mandarin-english* and *modern standard arabic-arabic* dialects. They used naive features like language model, language specific and morphological features.

Das and Gambäck [8], have performed language detection in Indian context *english-bengali* and *english- hindi*. They used SVM with linear kernel by using features like n-grams with weights, dictionary based, minimum edit distance based weight and word context features. The system proposed by Barman et al. [4] identified code-switching in *nepali-english* and *spanish-english*. The features include character n-grams, dictionary based features, length of words, capitalization and contextual information. In Bar and Dershowitz [2] the authors used a sequential classifier with character and word level features. The experiments are performed in a supervised fashion by using SVM with polynomial kernel of degree 2. They have used a window of 2 words before and after the word of interest as features. In Barman et al. [3] too followed a similar approach using SVM and CRF and their results emphasize the importance of contextual information that is obtained from dictionaries. Chittaranjan et al. [7], Lin et al. [18], Al-Badrashiny and Diab [1] have described a generic CRF based word labeling system to identify code-mixing. Mandal et al. [20] proposed a voting approach by training multiple classifiers instead of a single classifier. Ghosh et al. [11] described a CRF system for query word language identification. The features used are capitalization, word level context, special character, dictionary features.

The problem of word level language identification has been addressed by King and Abney [16] in a weakly supervised fashion. The character n-gram features are used in the system and it is found that 1 to 5 grams along with whole word features improve the accuracy.

Samih et al. [26] proposed LSTM (Long Short-Term Memory) neural network with CRF for language detection in code-switching using character n-grams and morphological as features. Jaech et al. [14] described C2V2L (Character to Vector to Language) model has two components. (i) char2vec uses CNN (convolutional neural network) to get word vector representation. (ii) The bi-directional LSTM recurrent neural network (RNN) maps the each word vector to language label. Xia [31] proposed CRF using a variant of word2vec to build word vectors for code-mixed language identification.

Most of the earlier works address the language identification in code-mixed text where the number of languages are two. Some of them address the code-mixing in single script and remaining address code-mixing in different scripts. The later problem is relatively easier while the complexity lies in distinguishing the tokens of different languages sharing the same script. We address this problem in the context of code-mixed text involving codes of 8 different Indian languages. We analysed the difficulty of the problem in two settings namely cross-validation and blind testset (which is more realistic). We proposed a two stage approach which consists of identifying the mixing language at the message level and applying a binary classifier to distinguish between *english* and *non-english* tokens.

3 APPROACH

3.1 Problem Definition

In this section we formalize mixed script language identification problem. Let $\mathcal{L}$ be the set of all languages $\{l_1, l_2, ..., l_n\}$. The word level language identification problem can be formalized as given a sentence as a word sequence $(w_1 w_2 ... w_s)$ predict the $(L_1 L_2 ... L_s)$ where $w_i \in \mathcal{V}$ for some vocabulary of words $\mathcal{V}$ and $L_i \in \mathcal{L}$. Though, all L_i can be a possibly different labels, the sentences of text being addressed in this problem are typically observed social media text and it can any of the two languages, typically with *english* as one of the mixing language. Our approach is generic to handle any number of these languages mixing in the same sentence.

This problem can be treated as version of sequence labeling problem with additional constraints where the number of mixing components can be limited. Ganchev et al. [10] proposed methods of enforcing such constraints on sequence labeling, making use of unlabeled data. Considering the restricted number of possible label combinations, we have chosen the path of mimicking what such an approach does by dividing the problem into two stages. First stage consists of identifying the mixing combinations of languages and

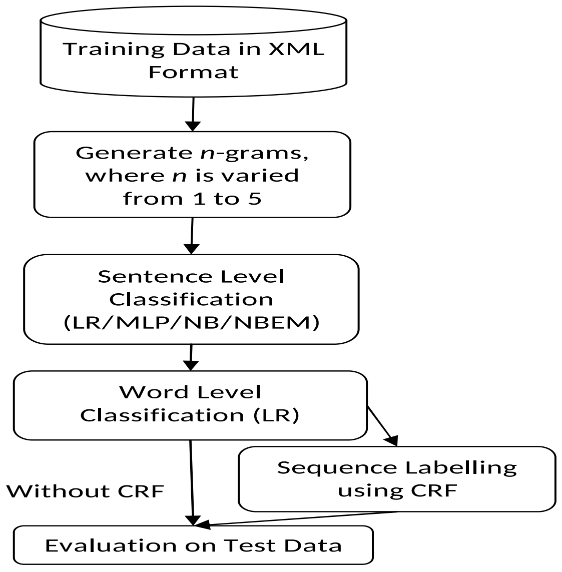
Training Data in XML Format
Generate n-grams, where n is varied from 1 to 5
Sentence Level Classification (LR/MLP/NB/NBEM)
Word Level Classification (LR)
Sequence Labelling using CRF
Without CRF
Evaluation on Test Data

Figure 1: Overview of the approach followed

second stage consists of using such a mixing combination specific binary classifier to distinguish between the mixing entities. Most of the earlier approaches consider differences in script to lead to language identification. We address a restricted version (which is quite relevant in the context of multi-cultural-lingual communities emerging with the social media platforms) of the problem where all the text is written in single script with the possibility of each of the words originating from different language. Such a two stage approach helped us analyzing the different aspects of the problem namely sentence level language mixture identification and word level language identification. Another benefit of such segregation is that the harder classification of first stage can be improved using techniques learnt from semi-supervised and un-supervised techniques.

The overall approach followed in this work is summarized in figure 1. The input text contained in xml/json is fed as input to our system. A subsequent pre-processing phase generates the features from the tokens of the text. In our approach, we utilise the character n-gram information of words. The second step consists of mixing language identification which utilises the character n-grams of the sentence. The third step consists of application of binary classifiers for identifying the word level language using appropriate binary classifier. The third step uses the mixing language identified in the previous stage. This is followed an optional fourth step, which takes as input the words and the word level language tags given in the previous stage and CRF labeller learnt from the training data to do the language tagging of the sentences, treating the word-sequence as an observation.

3.2 Word Level Classification

This work is carried out in the context of code-mixing involving *english* and some of the popular Indian languages namely *bengali*, *gujarati*, *hindi*, *kannada*, *marathi*, *malayali*, *tamil* and *telugu*. Most of the languages follow their own orthography when written in native script, but for the problem perspective all are written in Roman script itself. We use back transliterations of words from each of the languages in *bengali*, *gujarati*, *hindi*, *kannada*, *marathi*, *malayali*, *tamil* and *telugu* to train a basic word level classifier to distinguish between native and *english* words. We collected the frequent words from song lyrics similar to the lines of work in [12]. 5000 words from *english* word-list in Leipzig Corpus [1] and 5000 language specific words are used to build binary classifiers. We used character n-grams upto 5-grams as features for building the classifier. Logistic regression was used for classification. The details of the classifier are given in the next section. Dividing the dataset into training and test split of 80%-20%, we could get an accuracy of 95% for this task which is confirming the results reported in other works [13].

3.3 Language Identification at Sentence Level

As we are working with a setup where multiple language words are used in single sentence, the first stage of the problem consists of identifying the mixing language combination. Observing that most of the users use *english* as one of the mixing language we pose the problem of identifying mixing language combination as multi-class classification with 9 classes. 8 classes for each of the languages mentioned in the previous section with *english* as mixing language. Class 9 is used for identifying *english* sentences only. The multi-class classification problem can be solved using Naive Bayes (generative) and Logistic Regression (discriminative) approaches.

3.3.1 *Naive Bayes , Logistic Regression & MLP.*

Let x be an example expressed as feature vector in feature space $\mathcal{F}$. In the context of current classification problem, $\mathcal{F}$ represents all the character n-grams used in the feature representation. We used upto 5-grams as features. If y is the corresponding label of the example drawn from label space $\mathcal{Y}$, naive bayes classification maximizes the joint likelihood of (x, y) expressed as $p(x, y)$. Given a set of N training samples the objective of naive bayes classifier is given in equation (2).

$$p(x,y) = p(x|y)p(y) = p(y)\prod_{i=1}^{|\mathcal{F}|} p(x_i|y) \tag{1}$$

$$maximize \prod_{n=1}^{N} p(x^{(n)}, y^{(n)}) \tag{2}$$

Logistic regression (being discriminative) maximizes the class conditional distribution. The objective of logistic regression for multi-class classification is given in (5). (4) is normalizer of the probability expressed in (3). η_y represents the parameter vector for class y and '.' stands for the dot product between parameter vector and feature vector.

$$p(y|x) = e^{\eta_y . x}/Z \tag{3}$$

$$Z = \sum_{y'} e^{\eta_{y'} . x} \tag{4}$$

$$maximize \prod_{n=1}^{N} p(y^{(n)}|x^{(n)}) \tag{5}$$

[1] http://corpora.uni-leipzig.de/

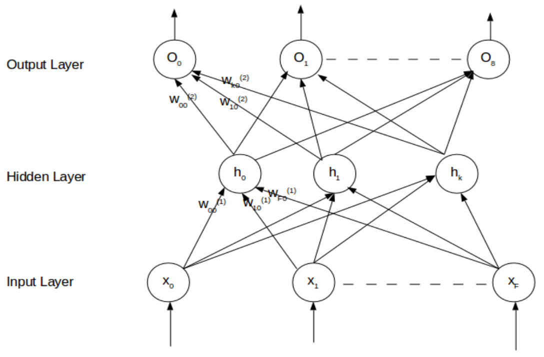

Figure 2: Multi-layer perceptron

Multi layer perceptron (MLP) trains a 3 layer neural network with one hidden layer, input layer with number of nodes equal to the number of input features and output layer with number of nodes equal to the number of output labels. Figure-2 depicts the network structure of MLP. The objective MLP is to minimize error on training data. Let the prediction function of MLP is $f : \mathcal{R}^{\mathcal{F}} \mapsto \mathcal{R}^{\mathcal{Y}}$. f is defined in equation (6). $A1$ and $A2$ are activation functions. $\mathbf{b}^1, \mathbf{b}^2$ and $\mathbf{W}^1, \mathbf{W}^2$ are bias vectors and weight matrices of respective layers. Back propagation is used for training the network to learn the parameters of the model. We use RELU(rectified linear units) and softmax as the activation functions $A1$ and $A2$ respectively.

$$f(\mathbf{x}) := A2(\mathbf{b}^2 + \mathbf{W}^2 * A1(\mathbf{b}^1 + \mathbf{W}^1 * \mathbf{x})) \tag{6}$$

3.4 Semi-Supervised Learning

Semi-supervised learning algorithms address scarcity of labeled examples. As mentioned earlier the multi-class classification problem which identifies sentence level language mixing class is an ideal case for semi-supervised learning. Unlabeled examples from the population of code-mixed text are abundantly available in social media.

We now formally define semi-supervised learning using expectation-maximization as introduced in [24] using naive bayes. Let $\mathcal{D}$ be the dataset containing the labeled $\mathcal{D}_{\mathcal{L}}$ and unlabeled examples $\mathcal{D}_{\mathcal{U}}$. The index of labeled examples in $\mathcal{D}$ ranges from 1..l (where l is the number of labeled examples) and unlabeled examples in the range (l+1)..(l+u) (where u is the number of unlabeled examples and N=l+u is the total number of examples). Expectation-maximization algorithm suggests the following approach for all the examples in $\mathcal{D}$.

The objective of this study is close the gaps in language identification using established machine learning techniques. There are approaches in the literature such as Jaech et al. [14] which try to integrate multiple aspects of the problem such as sequence labeling (used in the end) and character based n-gram approach (used at word level), we found that they do not work as the complete model needs to work as intended. It is also difficult to debug and fine-tune different aspects of the problem through side information such as semi-supervision (which are well studied in Machine Learning).

```
Data: Labeled Data $\mathcal{D}_L$ Unlabeled Data $\mathcal{D}_U$
Result: Model
initialize the model parameters $\theta^0$ from the labeled data $\mathcal{D}_L$ ;
while not convergence do
    E Step: Use the current parameters $\theta^t$ to learn
      $p(y^{(n)}|x^{(n)}) \forall n \in \mathcal{D}_{\mathcal{U}}$ ;
    M Step: Use the conditional distribution $p(y^{(n)}|x^{(n)})$
      to learn the model parameters $\theta^{t+1}$ ;
end
```

Algorithm 1: EM algorithm for semi-supervised learning

3.5 Sequence Labeling

In the language identification problem, as a final measure for improving accuracy we use sequence labeler to predict the sequence of language labels from the set of labels given by the word level binary classifier. This often improves labeling accuracy of words which are ambiguous for the binary classifier. Words which are shorter, often overlap with the other language leading wrong labeling. In such a scenario context of the word based on the label of previous/next word can help in resolving the ambiguity. The sequence labeling approach is solved using CRF. The objective of CRF is similar to that of logistic regression (5), with additional features and parameters encoding the dependency of successive labels in a sentence.

The CRF based sequence labeller models the maximization of joint conditional likelihood in equation (5) where $p(\mathbf{y_i}|\mathbf{x_i})$ is defined as in equation (7). In sequnce labeling models, $(\mathbf{x_i}, \mathbf{y_i})$ is the i'th example where $\mathbf{x_i}$ is a feature sequence of length T and $\mathbf{y_i}$ is corresponding label sequence. The difference between the numerator in equations (3) and (7) lies in features considered. CRF tries to model sequential depedencies, while simple multi-class logistic regression classifier disregards sequential dependecies. The feature vector $\mathbf{f}(\mathbf{x_i}, \mathbf{y_i})$ in equation (7) is similar to the feature vector in equation (5), encoding number of times a feature is associated with a label. The feature vector $\mathbf{g}(\mathbf{y_i})$ encodes the label sequence features or number of times a label combination appears in succession in the example $(\mathbf{x_i}, \mathbf{y_i})$. The number of features of $\mathbf{g}(\mathbf{y_i})$ vector is $|\mathcal{Y}|X|\mathcal{Y}|$ each encoding possible label bigrams. The denominator $Z(\mathbf{x_i})$ is normalizer which is evaluated over all possible label assignments $|\mathcal{Y}|^T$. Evalutaion of $Z(\mathbf{x_i})$ is efficiently done using forward-backward algorithm. μ and η are respective parameters associated with node features and edge features of CRF.

$$p(\mathbf{y_i}|\mathbf{x_i}) = \frac{exp(\mu^t.\mathbf{f}(\mathbf{x_i}, \mathbf{y_i}) + \eta^t \mathbf{g}(\mathbf{y_i}))}{Z(\mathbf{x_i})} \tag{7}$$

We refer to the tutorial by Sutton et al. [29] for a thorough treatment of CRF.

4 EXPERIMENTS

We have presented the results of our experiments on a dataset from FIRE-2015 shared task on Mixed Script information retrieval. This dataset contains the text collected from social media and each word tagged with one of the 9 language ids. numbers, digits and special symbols are labeled with "X". After preprocessing, we have considered 3,659 out of 3,699 instances for classification. In table 1, the fist column describes the number of sentences from each language, second column describes the number of tokens in total

Table 1: Summary of Dataset FIRE-2015

Language	Instances	Tokens	Lang Tokens
English	824 (676/148)	14827 (12625/2202)	22101(8031/4070)
Bengali	355 (215/140)	8074 (5902/2172)	4933 (3563/1370)
Gujarati	165 (149/16)	1482 (937/545)	1075 (890/185)
Hindi	617 (393/224)	14627 (11856/2771)	6051 (4458/1593)
Kannada	373 (272/101)	3728 (2735/993)	2279(1674/605)
Malayalam	151 (131/20)	2585 (1975/610)	1399 (1160/239)
Marathi	229 (200/29)	3318 (2696/622)	2414 (1960/454)
Tamil	342 (318/24)	6745 (5758/987)	3712 (3169/543)
Telugu	603 (525/78)	7877 (6812/1065)	7014 (6477/537)
Total	**3659**	**63262**	**50978**

sentences of each language and last column describes the number of tokens from each language in FIRE2015 dataset. The numbers in each of the columns are split into corresponding split of train/test counts in the parantheses. For example 8074 tokens of *bengali* are divided into 5902 tokens in the training data and 2172 tokens in the test data. The splitting is same as the one used in FIRE-shared task. We report the results of cross-validation and the performance on FIRE sharedtask splitting of the data.

The FIRE-2015 dataset is imbalanced with the presence of *english* as most frequent and gujarati as the least frequent. We train four training algorithms Naive Bayes(NB), Logistic Regression (MaxEnt), Multi Layer Perceptron (MLP) and Naive Bayes EM (NBEM). We used implementations of NB, MaxEnt and NBEM available in mallet [21]. We have used Keras a deep learning library for MLP implementation with theano as the backend. We used CRF++ [2] for the CRF implementation.

The performance of different algorithms is measured through F-score. We define F-score interms of precision and recall. Precision and Recall are measured for every language of interest and F-score is also calculated for every such language. The precision, recall and F-score are defined in equations in (10).

$$Precision(P) = \frac{\#CorrectPredictions}{TotalPredictions} \tag{8}$$

$$Recall(R) = \frac{\#CorrectPredictionsRecovered}{Totalnumberofactuallabels} \tag{9}$$

$$F-Score = \frac{2*P*R}{P+R} \tag{10}$$

The results in Table-2 shows the F-score averaged over 5 fold cross-validation. As the dataset is well-mixed we see that discriminative approaches fared better than generative NB method. Though we saw similar results for SVM, we report only two discriminative approaches tried through out our experimentation. The table-2 shows that performance of these classifiers on *english* is almost similar across all the classifiers, because of sufficient labeled information for this part of the classifier. Except for *bengali* and *hindi*, MLP performed much better compared to the rest. NBEM performed the best in these two cases.

When we analysed the errors, we observed that the errors are more when the length of words is lesser, which means that the

[2]https://taku910.github.io/crfpp/

Table 2: Strict F-measure of classifiers versus languages with 5 fold cross validation

Language	MaxEnt	NaiveBayes	NBEM	MLP
English	0.8665	0.8622	0.8662	**0.8674**
Bengali	0.8403	0.8259	**0.8575**	0.8516
Gujarati	0.7495	0.0108	0.5652	**0.8168**
Hindi	0.7735	0.7419	**0.7998**	0.7943
Kannada	0.8657	0.8155	0.8743	**0.8806**
Malayalam	0.8343	0.6230	0.8780	**0.8992**
Marathi	0.8423	0.7395	0.8643	**0.8789**
Tamil	0.9049	0.8729	0.9192	**0.9270**
Telugu	0.9144	0.8712	0.9109	**0.9252**

Table 3: Strict F-measure of classifiers + CRF versus languages with 5 fold cross validation

Language	MaxEnt	NaiveBayes	NBEM	MLP
English	**0.9169**	0.9114	0.9138	0.9149
Bengali	0.8788	0.8713	**0.8947**	0.8871
Gujarati	0.7776	0.5883	0.6030	**0.8513**
Hindi	0.8419	0.8122	**0.8596**	0.8554
Kannada	0.8771	0.8277	0.8868	**0.8920**
Malayalam	0.8405	0.6340	0.8836	**0.9060**
Marathi	0.8851	0.7932	0.9082	**0.9240**
Tamil	0.9284	0.8956	0.9426	**0.9508**
Telugu	0.9607	0.9207	0.9559	**0.9722**

word level classification is not being done correctly, though the mixing language is recognised correctly. When the length of the words is lesser, it should be the surrounding word labels which are detrimental in deciding the language id of the current word. We performed an experiment to understand the effectiveness of such an approach. We used a final sequence labeling algorithm based on CRF to verify this. The results are summarized in table 3. We can observe that the F-scores have improved across different languages. The improvement is highest in *english*, *hindi* and *telugu* (5-7%). The improvement is moderate in *gujarati*, *bengali* and *marathi* (3-4%), while it is least among *tamil*, *kannada* and *malayalam* (1-2%). We attribute this phenomenon to the longer average word-lengths in these languages which itself is sufficient for accurate word level tagging.

4.1 Results on blind test data

Table-4 and table-5 have presented the F1-Scores of 8 models on same dataset for the split of FIRE-shared task. In this experiment we have considered 2879 instances as train data and 780 instances as test data among the 3659 tuples of data samples. As it is meant for the shared-task, the distribution of training data and test data (terminology) differ significantly compared to the splits of the cross-validation folds. The details of the dataset w.r.t the train/test split are given in the table 1. The numbers in parantheses represents the train-test splitting of the total. While MLP fared better than other 3 learning approaches in the cross-validation setting, it was the semi-supervised setup which resulted in much improvement in accuracy of the first level multi-label classification. As shown in table-4 and table-5, the F-scores of NBEM are best in all the languages except for *gujarati* and *malayalam*. *Gujarati* sentences are strangely not detected by any of the sentence level classifiers being used in our system, which is the reason for the entire row being zero in tables 4 and 5. The ratio of language tags vs total tokens is on the lower side for *gujarati*. We further observed that among 185 language tokens of *gujarati* 84% of the tokens are below length 5 and do not carry the n-gram information sufficient enough to discriminate the *gujarati* sentences.

To understand the overall performance of each of the approaches tried, we summarize the average F-score. Unlike the average F-score reported in FIRE-sharedtask [27] which considers F-score of 'X' tags, we restrict our attention to language specific F-scores. For the same reason the average F-score is slightly on the lower scale compared to that of the reported values in the shared-task [27]. The average F-scores are summarized in the table 6. The average F-score for NB is 0.6064 while that of NBEM with CRF is 0.6668. We could observe that with CRF, the improvement over the our submission (3rd in average) results of FIRE-2015 shared task is around 10% in terms of F-score. Considering the X-tag F-scores are similar for the top-best submissions, we can see that improvement is around 6%.

Table 4: Strict F-measure of classifiers versus languages

Language	MaxEnt	NaiveBayes	NBEM	MLP
English	0.83	0.8343	**0.8362**	0.8287
Bengali	0.7988	0.8263	**0.8295**	0.7971
Gujarati	0	0	0	0
Hindi	0.6426	0.7063	**0.7429**	0.6462
Kannada	0.8171	0.7267	**0.8192**	0.8175
Malayalam	**0.8345**	0.6667	0.7807	0.8009
Marati	0.7486	0.7848	**0.8157**	0.8009
Tamil	0.8362	0.8291	**0.8942**	0.8688
Telugu	0.6131	0.6248	**0.7524**	0.6041

Table 5: Strict F-measure of classifiers + CRF versus languages

Language	MaxEnt	NaiveBayes	NBEM	MLP
English	**0.8728**	0.87	0.8699	0.8716
Bengali	0.8241	0.8556	**0.8615**	0.8205
Gujarati	0	0	0	0
Hindi	0.6913	0.746	**0.7856**	0.6992
Kannada	0.8227	0.7338	**0.8235**	0.8208
Malayalam	**0.8399**	0.7059	0.7791	0.8066
Marati	0.7856	0.8478	**0.8658**	0.8505
Tamil	0.8526	0.8377	**0.9132**	0.8849
Telugu	0.6045	0.5783	**0.7265**	0.6061

5 CONCLUSIONS

This work studies the problem of word-level language identification with machine learning approaches using character n-gram based features. We observe that a two stage approach of dealing with the word-level language identification leads to much more efficient approach as it helps in isolating the two problems of finding overall mixing language combination and then further applying corresponding classifier to distinguish the combination specific mixing languages. We further apply CRF based sequence labeling on the result to improve the classification accuracy. The segregation of the problem into respective individual sub-problems helps in applying novel machine learning algorithms suitable for respective tasks. We also found that using semi-supervised approach leads to better accuracy for the first level.

Table 6: Average F-measure of classifiers

Method	Average F-Score
MaxEnt	0.6166
NB	0.6064
NBEM	**0.6457**
MLP	0.6202
MaxEnt + CRF	0.6381
NB + CRF	0.6289
NBEM + CRF	**0.6668**
MLP + CRF	0.6453

ACKNOWLEDGMENTS

We thank IDRBT for supporting us to carry-out this work.

REFERENCES

[1] Mohamed Al-Badrashiny and Mona Diab. 2016. The George Washington University System for the Code-Switching Workshop Shared Task 2016. *EMNLP 2016* (2016), 108.
[2] Kfir Bar and Nachum Dershowitz. 2014. The Tel Aviv university system for the code-switching workshop shared task. *EMNLP 2014* (2014), 139.
[3] Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. *EMNLP 2014* 13 (2014).
[4] Utsab Barman, Joachim Wagner, Grzegorz Chrupała, and Jennifer Foster. 2014. Dcu-uvt: Word-level language classification with code-mixed data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching. EMNLP 2014, Conference on Empirical Methods in Natural Language Processing*. 127–132.
[5] Marine Carpuat. 2014. Mixed-language and code-switching in the canadian hansard. *EMNLP 2014* (2014), 107.
[6] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. 2010. *Semi-Supervised Learning* (1st ed.). The MIT Press.
[7] Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury. 2014. Word-level language identification using crf: Code-switching shared task report of msr india system. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*. 73–79.
[8] Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing, Goa, India*. Citeseer, 169–178.
[9] Ramy Eskander, Mohamed Al-Badrashiny, Nizar Habash, and Owen Rambow. 2014. Foreign words and the automatic processing of Arabic social media text written in Roman script. *EMNLP 2014* (2014), 1.
[10] Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior Regularization for Structured Latent Variable Models. *J. Mach. Learn. Res.* 11 (Aug. 2010), 2001–2049. http://dl.acm.org/citation.cfm?id=1756006.1859918
[11] Satanu Ghosh, Souvick Ghosh, and Dipankar Das. 2016. Labeling of Query Words using Conditional Random Field. *arXiv preprint arXiv:1607.08883* (2016).
[12] Kanika Gupta, Monojit Choudhury, and Kalika Bali. 2012. Mining Hindi-English Transliteration Pairs from Online Hindi Lyrics.. In *LREC*. European Language Resources Association (ELRA), 2459–2465.
[13] Parth Gupta, Kalika Bali, Rafael E Banchs, Monojit Choudhury, and Paolo Rosso. 2014. Query expansion for mixed-script information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 677–686.

[14] Aaron Jaech, George Mulcaire, Shobhit Hathi, Mari Ostendorf, and Noah A Smith. 2016. A neural model for language identification in code-switched tweets. *EMNLP 2016* (2016), 60.
[15] Naman Jain, IIIT-H LTRC, and Riyaz Ahmad Bhat. 2014. Language identification in code-switching scenario. *EMNLP 2014* (2014), 87.
[16] Ben King and Steven P Abney. 2013. Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods.. In *HLT-NAACL*. 1110–1119.
[17] Yanen Li, Huizhong Duan, and ChengXiang Zhai. 2012. A generalized hidden markov model with discriminative training for query spelling correction. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 611–620.
[18] Chu-Cheng Lin, Waleed Ammar, Lori Levin, and Chris Dyer. 2014. The cmu submission for the shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*. 80–86.
[19] Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*. Association for Computational Linguistics, 25–30.
[20] Soumik Mandal, Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. 2015. Adaptive Voting in Multiple Classifier Systems for Word Level Language Identification.. In *FIRE Workshops*. 47–50.
[21] Andrew McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. (2002). http://mallet.cs.umass.edu
[22] Paul McNamee. 2005. Language Identification: A Solved Problem Suitable for Undergraduate Instruction. *J. Comput. Sci. Coll.* 20, 3 (Feb. 2005), 94–101. http://dl.acm.org/citation.cfm?id=1040196.1040208
[23] Pieter Muysken. 2000. *Bilingual speech: A typology of code-mixing*. Vol. 11. Cambridge University Press.
[24] Kamal Paul Nigam. 2001. *Using Unlabeled Data to Improve Text Classification*. Ph.D. Dissertation. Massachusetts Institute of Technology.
[25] Shana Poplack. 1980. Sometimes IâĂŹll start a sentence in Spanish Y TERMINO EN ESPAÑOL: toward a typology of code-switching1. *Linguistics* 18, 7-8 (1980), 581–618.
[26] Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual Code-switching Identification via LSTM Recurrent Neural Networks. *EMNLP 2016* (2016), 50.
[27] Royal Sequiera, Monojit Choudhury, Parth Gupta, Paolo Rosso, Shubham Kumar, Somnath Banerjee, Sudip Kumar Naskar, Sivaji Bandyopadhyay, Gokul Chittaranjan, Amitava Das, et al. 2015. Overview of FIRE-2015 Shared Task on Mixed Script Information Retrieval.. In *FIRE Workshops*, Vol. 1587. 19–25.
[28] Kalika Bali Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. âĂIJI am borrowing ya mixing?âĂİ An Analysis of English-Hindi Code Mixing in Facebook. *EMNLP 2014* (2014), 116.
[29] Charles Sutton, Andrew McCallum, et al. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning* 4, 4 (2012), 267–373.
[30] Liling Tan, Marcos Zampieri, Nikola Ljubesic, and Jorg Tiedemann. 2014. Merging Comparable Data Sources for the Discrimination of Similar Languages: The DSL Corpus Collection. In *9th International Conference on Language Resources and Evaluation (LREC), MAY 26-31, 2014, Reykjavik, ICELAND*.
[31] Meng Xuan Xia. 2016. Codeswitching language identification using subword information enriched word vectors. *EMNLP 2016* (2016), 132.