Team WebArch at FIRE-2018 Track on Indian Native Language Identification

Aman Gupta

SRM University, Kattankulathur, Chennai, Tamil Nadu, India

Abstract. Native Language Identification (NLI) is the task which involves identification of native language (L1) of an individual based on his/her language production in a learned language (L2). It is basically a classification task where we are classifying L1 into a number of different languages. In this task I have to identify an individual's native language (L1) among the following six Indian languages: Bengali, Hindi, Kannada, Malayalam, Tamil, and Telugu using their Facebook comments written in English language (L2). In this paper I propose to use machine learning models such as classification models together with N-grams as features and Tfidf as vectorizer.

Keywords: Native Language Identification · Natural Language Processing · Classification.

1 Introduction

Native Language Identification (NLI) is the task of classifying the native language (L1) of an individual into given different languages based on his/her writing in another language (L2). NLI tasks involves identifying language use patterns that are common to certain groups of speakers that share the same native language. The native language of an individual influences the usage of words as well the errors that a person makes when writing in another language. The task is usually considered as a classification problem where a machine learning algorithm is trained in a supervised manner which can then be used for predicting the native language of user text.

Predicting the native language of a writer has applications in different fields. It can be used for authorship identification, forensic analysis, tracing linguistic influence in potentially multi-author texts and naturally to support Second Language Acquisition research. In the field of cyber security, NLI can be used to determine the native language of an author of a suspicious or threatening text. In the field of academics NLI can be used for educational applications such as developing grammatical error correction systems which can personalize their feedback and model performance to the native language of the user.

In my research here I have used classification models such as Logistic Regression, Linear SVC, Naive Bayes to name a few, together with features such as N-Grams both at character level and word level to find the best working model which can efficiently classify L1 of a person to different sets of languages.

2 Related Work

NLI research has mostly been focused on texts where both lexical and syntactic features were used. Models formed try to extract patterns that speakers with different native language will have in terms of different misspellings, mispronunciations or usage frequency of particular words. Also some languages have specific linguistic styles, like Japanese is much more formal in nature while French and Spanish are way more romantic in nature because of their gentler vocabulary whereas Russian and German can be classified as harsh because of their string vocabulary. Kumar et al.[1] (2017) published an overview of FIRE-2017 which similar to this was based on Native Language Identification using comments of individuals on social networking sites. Malmasi et al.[2] published a report on Native Language Identification Shared Task which depicted various approaches taken by participants for solving Native Language Identification task. Malmasi and Dras[4] tested a range of linear classifiers, and observed that state of the art results was achieved by an ensemble model in 2017. The features they used were simple unigrams, bigrams, and character n-grams. They also found that character level features generally outperform word level features for NLI. Tsur and Rappoport[6] also (2007) achieved an accuracy of 66% by using only character bi-grams. Besides these Swanson and Charniak[7] uses a bit different approach of using Tree Substitution Grammars (TSGs). Wong and Dras[8] also explored production rules from two parsers in 2011.

3 Task Description and Data

Dataset provided by task organizers contains information collected from English speakers of six different native Indian languages namely Tamil, Telugu, Kannada, Malayalam, Bengali and Hindi. Data was collected through social networking site Facebook. The distribution of class and training instances can be seen in Table 1.

Language	**Training Instances**
Bengali	202
Hindi	211
Kannada	203
Malayalam	200
Tamil	207
Telugu	210

Table 1: Distribution of Training Data

Table 1. shows number of instances for each of the six languages is roughly the same. Task was to predict L1 of a candidate based on tweets posted by an individual on social networking site written in L2.

4 Proposed Technique

I have tried to model this task as a classification task. I have divided the given dataset into three parts Training set (75%) , Testing set(12.5%) and Validation set(12.5%).

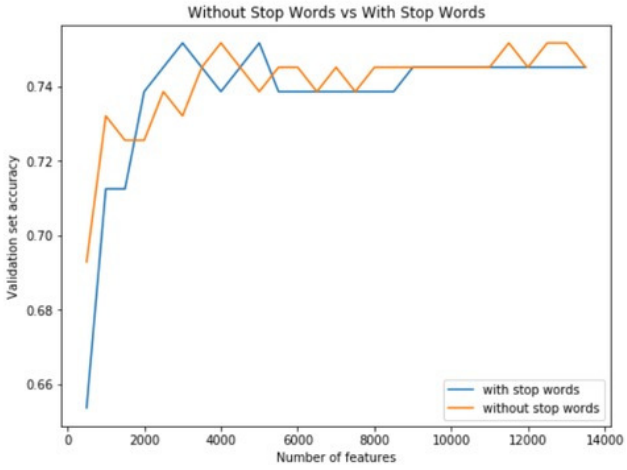
Languages	**Training**	**Test**	**Validation**
Bengali	143	28	22
Hindi	160	30	21
Kannada	156	19	28
Malayalam	149	16	29
Tamil	152	29	26
Telugu	153	30	27
Total	913	152	153

Table 2: Distribution of Data into Training, Validation and Test Sets

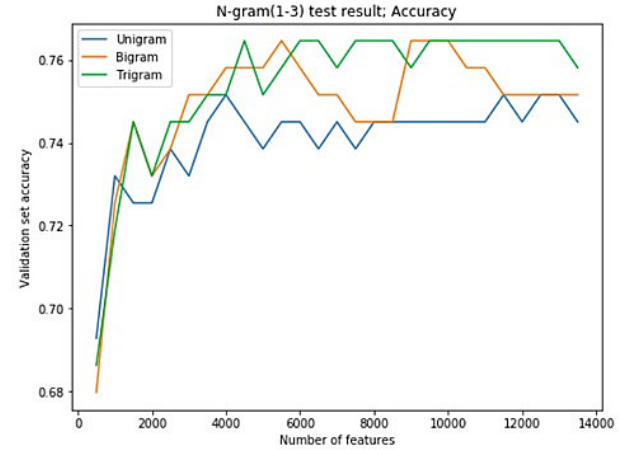
Validation set was used to determine several hyper parameter values. For determining whether to use stop words or not I used Count Vectorizer as a sample vectorizer to calculate token counts, Logistic Regression as a sample classification model. Two classification models were trained one " with stop words" other " without stop words". Both were trained using training set data while accuracy was calculated on validation set. I used Python's NLTK stop words as sample. The results from both models were plotted (see Fig1.a). Validation Set Accuracy being represented on Y-axis and Number of features(maximum no of words in vocabulary) on X-axis.

Accuracy for category "without stop words" was found to be higher when number of features considered are large.Similar to this Validation Set was also used to determine which N-gram are giving the best results.Unigrams, Bigrams, Trigrams were under consideration. So I trained three sample Logistic regression model one for every N-gram using training set data. Results from all three models were plotted(see Fig1.b).Validation Set Accuracy being represented on Y-axis and Number of features (maximum no of words in vocabulary) on X-axis.

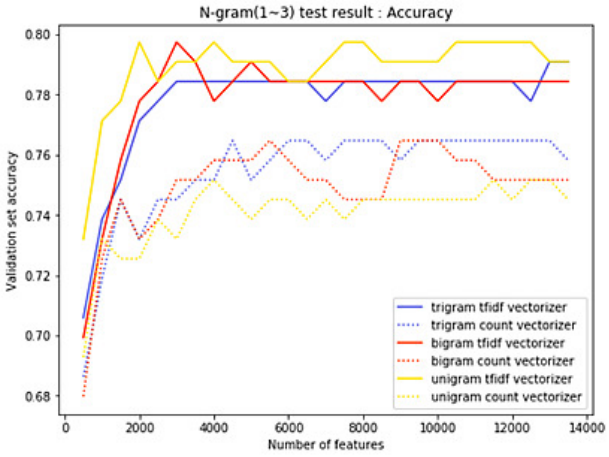
Accuracy for Unigrams were found to be highest when large number of features are considered. Hence it is Unigrams which are able to best capture inherent features of a language. Lastly I also determined which vectorizer to use. In my research I considered TfIdf Vectorizer and Count Vectorizer as two different candidates. Count Vectorizer basically converts a collection of text documents to a matrix of token counts whereas TfIdf Vectorizer converts a collection of raw documents to a matrix of Tfidf features in simpler terms it is basically product of term frequency(no of times a particular word appears in a single document) and inverse document frequency(log of number of docs in your corpus divided by the number of docs in which this term appears). In total six sample Logis-



(a) StopWords Or Not



(b) N-grams



(c) Vectorizer

Fig. 1: Validation Set Results

tic regression models were trained using training set data while accuracy was calculated using validation set.

Accuracy for Tfidf vectorizer for Unigrams were found to be highest. Now after determining the hyper parameter values I considered several classification models. Each model was trained on training set while accuracy was calculated using Validation set and compared with null accuracy (accuracy calculated from model which always predicts the label which appeared maximum number of times in training set). Table 3 summarizes the results.

Classifier	**Accuracy(%)**	**With Null Accuracy(%)**	**Time(s)**
Logistic Regression	79.74	62.21	0.26
Linear SVC	82.35	64.83	0.26
Linear SVC(L1 selection)	74.51	56.99	0.47
Multinomial NB	77.12	59.60	0.19
Bernoulli NB	55.56	38.03	0.19
Ridge Classifier	82.35	64.83	0.37
SGD Classifier	77.78	60.25	0.26
AdaBoost	49.67	32.15	0.83
Perceptron	76.47	58.95	0.22
Passive-Aggressive	83.01	65.48	0.26
Nearest Centroid	79.08	61.56	0.22

Table 3: Accuracy of different classifiers.

Based on the accuracy on Validation Set Passive-Aggressive classifier showed the best results. Accuracy on Test dataset for Passive-Aggressive classifier was found to be 82.89%.

5 Test and Results

Test Dataset was released on much later date and our final model was tested on it. Test Dataset provided by organizers consisted Facebook comments for various Indian Languages.

On test set1 accuracy achieved was 41.4% while accuracy on test set2 which was released on a later date accuracy achieved was 31.9%. Accuracy is much lower than that achieved during training due to lack of large training dataset due to which model was not able to extract useful patterns of different languages efficiently.

Acknowledgements

I would like to express my special thanks to Soumil Mandal for his guidance and constant supervision as well as providing necessary information. Also I would like to extend my gratitude towards Team WebArch for providing this opportunity.

References

1. Anand Kumar, M., Barathi Ganesh, H.B., Singh, S., Soman, K.P., Rosso, P. *Overview of the INLI PAN at FIRE-2017 track on Indian native language identification (2017)* CEUR Workshop Proceedings, 2036, pp. 99-105.
2. Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano and Yao Qian. *A Report on the 2017 Native Language Identification Shared Task.*
3. Shervin Malmasi et al. 2016. *Native language identification: explorations and applications.*
4. Shervin Malmasi and Mark Dras. 2017. *Native Language Identification using Stacked Generalization.*
5. Scott Jarvis, Yves Bestgen, and Steve Pepper. 2013.*Maximizing Classification Accuracy in Native Language Identification.*
6. Oren Tsur and Ari Rappoport. 2007. *Using Classifier Features for Studying the Effect of Native Language on the Choice of Written Second Language Words.*
7. Benjamin Swanson and Eugene Charniak. 2012. *Native Language Detection with Tree Substitution Grammars.*
8. Sze-Meng Jojo Wong and Mark Dras. 2011. *Exploiting Parse Structures for Native Language Identification.*
¡aman304gupta@gmail.com¿ 2018-10-25T14:49:57.338Z: