Mangalore-University@INLI-FIRE-2017: Indian Native Language Identification using Support Vector Machines and Ensemble Approach

Hamada A. Nayel
Department of Computer Science
Mangalore University, Mangalore-574199, India
Benha University, Benha-13518, Egypt
hamada.ali@fci.bu.edu.eg

H. L. Shashirekha
Department of Computer Science
Mangalore University, Mangalore-574199
Karnataka, India
hlsrekha@gmail.com

ABSTRACT

This paper describes the systems submitted by our team for Indian Native Language Identification (INLI) task held in conjunction with FIRE 2017. Native Language Identification (NLI) is an important task that has different applications in different areas such as social-media analysis, authorship identification, second language acquisition and forensic investigation. We submitted two systems using Support Vector Machine (SVM) and Ensemble Classifier based on three different classifiers representing the comments (data) as vector space model for both systems and achieved accuracy of 47.60% and 47.30% respectively and secured second rank over all submissions for the task.

CCS CONCEPTS

• **Information systems** → *Web and social media search*; *Multilingual and cross-lingual retrieval*; • **Computing methodologies** → **Language resources**;

KEYWORDS

Support Vector Machines, Ensemble Learning, Native Languages Identification, Word Vector Space

1 INTRODUCTION

Native Language Identification (NLI) aims at identifying the native language (L1) of users writing in another or later learned language or speech (L2). NLI is an important task that has many applications in different areas such as social-media analysis, authorship identification, second language acquisition and forensic investigation. In forensic analysis [7], NLI helps to glean information about the discriminant L1 cues in an anonymous text. Second Language Acquisition (SLA) [12] studies the transfer effects from the native languages on later learned language. In education, automatic correction of grammatical errors is an important application of NLI [14]. NLI can be used as a feature in authorship identification task [6], which aims at assigning a text to one of the predefined list of authors. Authorship identification is used for terrorists communications investigation [1] and digital crime investigation [4].

Supervised approaches using machine learning algorithms have been used for NLI by many researchers. Jarvis et al. [9], used SVM classification algorithm to create a model for NLI and reported an accuracy of 83.6%. They used features such as n-grams of words, Part-of-Speech (PoS) tags and lemmas. Combining multiple classifier systems to enhance the final output, such as ensemble classifier was used for NLI by Tetreault et al. [15]. Bykh and Meurers [3] applied a tuned and optimized ensemble classifier on NLI 2013 shared task dataset and achieved an accuracy of 84.82%.

2 TASK DESCRIPTION

Given a comment $\mathcal{I} = <w_1, w_2, \ldots, w_N>$ where each w_i, $i = 1..n$ is either an English language word or a word of native language written in English (or transliterated to English language) for an individual social media user, the objective of the task is to identify the native language of the user. The comment may include English words in addition to the words of any one native language written in English. The task considers six Indian languages, namely Tamil (TA), Hindi (HI), Kannada (KA), Malayalam (MA), Bengali (BE) and Telugu (TE). Considering the languages as a set of classes $C = \{TA, HI, KA, MA, BE, TE\}$ and comments as individual instances $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_n\}$ we have formulated the task as a classification problem that assigns one of the six predefined classes of C to a new unlabelled instance $\mathcal{I}_u$.

3 DATASET

The data sets provided for this task are a collection of comments from different regional newspaper's facebook pages during April-2017 to July-2017. Training and test sets contain 1233 and 783 files respectively. Each training and testing file consists of a set of comments. Table 1 shows a brief statistics about training set.

Table 1: Training set statistics

Language	# of comments	Ratio
TA	207	16.79%
HI	211	17.11%
KA	203	16.46%
MA	200	16.22%
BE	202	16.38%
TE	210	17.03%
Total	1233	100%

4 SYSTEM DESCRIPTION

In this section, we will describe the two systems proposed for Indian Native Language Identification (**INLI**) [10] task submissions. The general frame work of classifier for both systems is shown in figure 1. First phase of our systems is data preprocessing, also known

as corpus cleaning. This phase is important where we exclude non-informative tokens and phrases. Second phase comprises of constructing vector space model for the comments (input data). These two phases are common for both the systems. The next phase is creating a model using a machine learning algorithm. Support Vector Machine (SVM) and Ensemble learning are used for the first and second submission respectively. Details of each phase is given below.

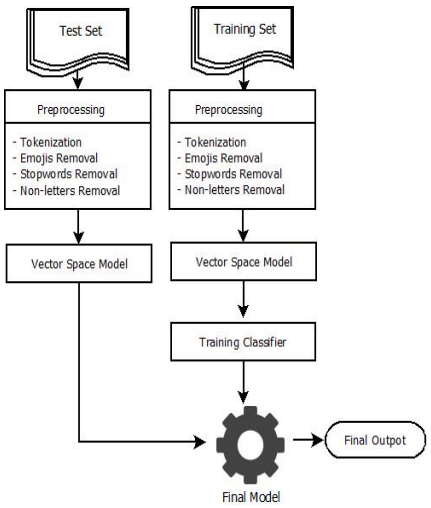

Figure 1: Framework of classifier

4.1 Pre-processing

In this phase, we tokenized each comment $\mathcal{I}_j$ into a set of words or tokens and removed uninformative tokens as follows to get bag of tokens:-

- ***Emoji removal***
  Emoji is a small image used as a visual presentation to express emotion. The first step in removing unrelated information is to remove Emojis as they are not important for the identification of native language.
- ***Special characters and digits***
  Digits and special characters such as #, %, ... are the characters which appear frequently in the comments of all the languages. As such characters do not contribute to the identification of native language they are removed.
- ***Modified stop words***
  Stop words are the words which appear frequently and do not contribute to the identification of native language. Hence, to remove stop words we used a union of different stop words lists, namely,
  (1) stop words list extracted from `nltk.corpus`[1] package.
  (2) stop words list extracted from `stop_words`[2] package.
  (3) Manually written stop words. (The complete list of manually written stop words is given in Appendix A)

[1]www.nltk.org/nltk_data/
[2]pypi.python.org/pypi/stop-words

4.2 Constructing Vector Space Model

After preprocessing, the comments will be represented as vector space model. If $<t_1, t_2, \ldots, t_k>$ are the unique tokens/terms in a comment $\mathcal{I}_j$, the vector space model for the comment $\mathcal{I}_j$ will be represented as $<w_{j1}, w_{j2}, \ldots, w_{jk}>$ where w_{ji} is the weight of the token/term t_i in comment $\mathcal{I}_j$. For term weights, we used Term Frequency/Inverse Document Frequency (TF/IDF) calculated as follows:-

$$t_j = tf_j * \log\left(\frac{N+1}{df_j+1}\right)$$

where tf_j is the total number of occurrences of term t_j in the current comment, df_j is the number of comments in which the token/term t_j occurs and N is the total number of comments.

4.3 Model Construction for First Submission using SVM

SVM is a binary classifier which creates a hyperplane that discriminates between the two classes [5]. SVM can be extended to multi-class problems by creating several binary SVMs and combining them using a one-vs-rest method or one-vs-one method [8].

We implemented a six class SVM corresponding to six classes TA, HI, KA, MA, BE and TE, as per the framework shown in figure 1 for comment identification using Stochastic Gradient Descent (SGD) for optimizing the parameters of SVM model. SGD algorithm updates the value of parameter θ of the objective function $W(\theta)$ as

$$\theta = \theta - \eta \nabla_\theta E\left[W(\theta)\right]$$

where η is step size and $E[W(\theta)]$ is the cost function.

4.4 Model Construction for Second Submission using Ensemble Approach

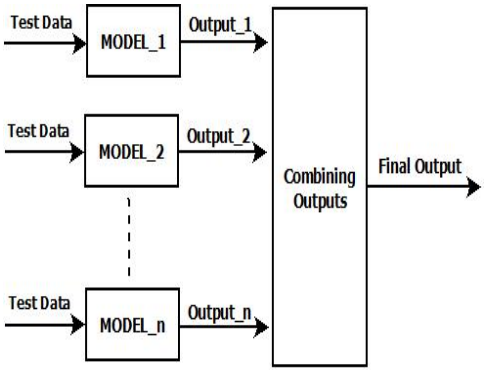

Figure 2: Framework of Ensemble approach

Ensemble learning is a classification technique, which uses a set of different heterogenous and diverse classifiers as base classifiers and combines the output of them in different approaches to get the final output [13]. Ensemble technique tries to overcome the weakness of some classifiers using the strength of other classifiers. Figure 2 shows the framework of ensemble learning.

We have used 3 base classifiers, namely, multinomial Bayes, SVM and random forest tree classifiers and combined the results

by weighted voting. Multinomial Bayes classifier is an instance of Naive Bayes classifier that captures word frequency information in documents [11]. Random forests classifier is a supervised classifier which comprises of multiple decision trees and each tree depends on independently sampled random vector [2]. The base classifiers are designed as per the framework shown in figure 1.

5 PERFORMANCE EVALUATION

Performance evaluation of INLI task is measured as the accuracy of the system in addition to class-wise accuracy which is calculated using Precision (**P**), Recall (**R**) and **F1** measure [3] . For each class, **P** is the measure of the number of comments correctly classified over the total number of comments that system classified as same class. **R** is the measure of the number of comments correctly classified over the actual number of comments of the class. **F1** measure is the harmonic mean of **P** and **R**, which can be calculate as follow:-

$$F1 = \frac{2 * P * R}{P + R}$$

6 RESULTS AND DISCUSSION

The class wise accuracy of first submission using SVM based on SGD algorithm to determine the parameters of the model is shown in Table 2 in terms of **P**, **R** and **F1** measure. The overall accuracy of this submission is 47.60% and it ranks second among all the submissions.

Table 2: Results of SVM classifier based submission

Class	P	R	F1
BE	54.00%	84.90%	66.00%
HI	60.00%	7.20%	12.80%
KA	40.40%	54.10%	46.20%
MA	42.70%	66.30%	51.90%
TA	58.00%	58.00%	58.00%
TE	32.50%	48.10%	38.80%
Overall Accuracy		47.60%	

Table 3 shows the performance evaluation of the second submission where we used Ensemble approach to combine output of different models. Overall accuracy of this submission is 47.30% and it ranks third among all the submissions.

Table 3: Results of Ensemble classifier based submission

Class	P	R	F1
BE	56.50%	79.50%	66.10%
HI	60.70%	6.80%	12.20%
KA	38.40%	58.10%	46.20%
MA	40.40%	70.70%	51.40%
TA	58.00%	58.00%	58.00%
TE	32.80%	49.40%	39.40%
Overall Accuracy		47.30%	

[3]http://www.nltk.org/_modules/nltk/metrics/scores.html

We used 10-fold cross-validation technique while training both classifiers, the cross validation accuracy of both submissions is given in Table 4.

Table 4: 10-fold cross-validation accuracy for both submissions

Submission 1	Submission 2
88.09%	87.30%
84.80%	84.80%
90.32%	90.32%
91.06%	91.06%
89.43%	86.18%
79.68%	80.49%
86.18%	90.24%
88.52%	89.34%
90.98%	90.16%
89.34%	91.80%
Mean = 87.84% STD = 3.32	Mean = 88.17% STD = 3.33

Results of both submissions illustrates that the performance of identifying Hindi is the worst. The reason may be most of the other languages' natives have knowledge of Hindi. Our systems depend essentially on the effective words for each language.

7 CONCLUSION

In this work, SVM and Ensemble classifier have been used for INLI. SVM outperforms the Ensemble classifier which combines different three classifiers. Our Support Vector Machine (SVM) submission secured second rank respectively over all submissions for the task.

A COMPLETE LIST OF MANUALLY WRITTEN STOP WORDS

The following is the full list of stopwords used in our system:-

```
{ a, about, above, across, after, afterwards, again,
against, all, almost, alone, along, already, also,
although, always, am, among, amongst, amoungst, amount,
an, and, another, any, anyhow, anyone, anything, anyway,
anywhere, are, around, as, at, back, be, became, because,
become, becomes, becoming, been, before, beforehand,
behind, being, below, beside, besides, between, beyond,
bill, both, bottom, but, by, call, can, cannot, cant,
co, con, could, couldnt, cry, de, describe, detail, do,
done, down, due, down, due, during, each, eg, eight,
either, eleven, else, elsewhere, empty, enough, etc,
even, ever, every, everyone, everything, everywhere,
except, few, fifteen, fifty, fill, find, fire, first,
five, for, former, formerly, forty, found, four, from,
front, full, further, get, give, go, had, has, hasnt,
have, he, hence, her, here, hereafter, hereby, herein,
hereupon, hers, herself, him, himself, his, how, however,
hundred, i, ie, if, in, inc, indeed, interest, into,
is, it, its, itself, keep, last, latter, latterly,
least, less, ltd, made, many, may, me, meanwhile,
```

might, mill, mine, more, moreover, most, mostly, move, much, must, my, myself, name, namely, neither, never, nevertheless, next, nine, no, nobody, none, noone, nor, not, nothing, now, nowhere, of, off, often, on, once, one, only, onto, or, other, others, otherwise, our, ours, ourselves, out, over, own, part, per, perhaps, please, put, rather, re, same, see, seem, seemed, seeming, seems, serious, several, she, should, show, side, since, sincere, six, sixty, so, some, somehow, someone, something, sometime, sometimes, somewhere, still, such, system, take, ten, than, that, the, their, them, themselves, then, thence, there, thereafter, thereby, therefore, therein, thereupon, these, they, thick, thin, third, this, those, though, three, through, throughout, thru, thus, to, together, too, top, toward, towards, twelve, twenty, two, un, under, until, up, upon, us, very, via, was, we, well, were, what, whatever, when, whence, whenever, where, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, whoever, whole, whom, whose, why, will, with, within, without, would, yet, you, your, yours, yourself, yourselves }

REFERENCES

[1] Ahmed Abbasi and Hsinchun Chen. 2005. Applying Authorship Analysis to Extremist-Group Web Forum Messages. *IEEE Intelligent Systems* 20, 5 (Sept. 2005), 67–75. https://doi.org/10.1109/MIS.2005.81

[2] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (01 Oct 2001), 5–32. https://doi.org/10.1023/A:1010933404324

[3] Serhiy Bykh and Detmar Meurers. 2014. Exploring Syntactic Features for Native Language Identification: A Variationist Perspective on Feature Encoding and Ensemble Optimization. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, 1962–1973. http://aclanthology.coli.uni-saarland.de/pdf/C/C14/C14-1185.pdf

[4] Carole E Chaski. 2005. WhoâĂŹs at the keyboard? Authorship attribution in digital evidence investigations. *International journal of digital evidence* 4, 1 (2005), 1–13.

[5] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[6] Dominique Estival, Tanja Gaustad, Son Bao Pham, Will Radford, and Ben Hutchinson. 2007. Author profiling for English emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*. 263–272.

[7] John Gibbons. 2003. *Forensic linguistics: An introduction to language in the justice system*. Wiley-Blackwell.

[8] Chih-Wei Hsu and Chih-Jen Lin. 2002. A Comparison of Methods for Multiclass Support Vector Machines. *Trans. Neur. Netw.* 13, 2 (March 2002), 415–425. https://doi.org/10.1109/72.991427

[9] Scott Jarvis, Yves Bestgen, and Steve Pepper. 2013. Maximizing Classification Accuracy in Native Language Identification. (2013), 111–118 pages. http://aclanthology.coli.uni-saarland.de/pdf/W/W13/W13-1714.pdf

[10] Anand Kumar M, Barathi Ganesh HB, Shivkaran S, Soman K P, and Paolo Rosso. 2017. Overview of the INLI PAN at FIRE-2017 Track on Indian Native Language Identification. In *Notebook Papers of FIRE 2017, FIRE-2017*. Bangalore, India, December 8-10, CEUR Workshop Proceedings.

[11] Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*. The COLING 2012 Organizing Committee, 41–48.

[12] Lourdes Ortega. 2009. *Understanding Second Language Acquisition*. Hodder Education, Oxford.

[13] R. Polikar. 2006. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine* 6, 3 (Third 2006), 21–45. https://doi.org/10.1109/MCAS.2006.1688199

[14] Alla Rozovskaya and Dan Roth. 2011. Algorithm Selection and Model Adaptation for ESL Correction Tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 924–933. http://www.aclweb.org/anthology/P11-1093

[15] Joel Tetreault, Daniel Blanchard, Aoife Cahill, and Martin Chodorow. 2012. Native Tongues, Lost and Found: Resources and Empirical Evaluations in Native Language Identification. In *Proceedings of COLING 2012*. The COLING 2012 Organizing Committee, 2585–2602. http://aclanthology.coli.uni-saarland.de/pdf/C/C12/C12-1158.pdf