Corplab INLI@FIRE-2018: Identification of Indian Native Language using Pairwise Coupling

Soumik Mondal, Athul Harilal and Alexander Binder

Singapore University of Technology and Design (SUTD)
8 Somapah Road, Singapore 487372
{mondal_soumik, athul_harilal, alexander_binder}@sutd.edu.sg

Abstract. We are going to describe the techniques and methodology used during the implementation of Indian Native Language Identification (INLI) that was organized at FIRE 2018. Native Language Identification (NLI) is a process of identifying the native language of a writer by analyzing their text written in another language, which is in this case English. The following six different Indian languages were considered in this task: Bengali, Hindi, Kannada, Malayalam, Tamil, and Telugu. We have used state of the art TF-IDF feature vectors with linear-SVM as a classifier in our analysis. The classifier has been used with three different strategies (*i.e.* One-vs-the-rest and Pairwise Coupling strategies as described in [7]) and achieved an accuracy of 42.1% for test TestSet-1 and 31.8% for test TestSet-2.

Keywords: Native Language Identification · Linear SVM · TF-IDF Feature · One Vs Rest Classifier · Pairwise Coupling.

1 Introduction

Study on the influence of native language while using a second language has been studied since 1950s in the linguistic literature domain. This has motivated research in NLI that aims to automatically identify the native language of a user by analyzing the text written in another language. NLI works on the assumption that the user's linguistic background will lead them to use native language (mother tongue) phrases/styles more frequently in their learned languages.

NLI is useful in different areas such as cyber-forensic, authorship identification, analysis on social media, and second language acquisition. NLI helps to identify the author's native language by analyzing text from web-blogs to monitor terrorist communications and for digital crime investigation [1, 9].

NLI is a nontrivial pattern recognition challenge especially with a small corpus for training which has several useful applications. Therefore it is gaining popularity and competitions are being organized in various conferences/events in recent years [11, 5]. Most commonly used features for NLI tasks are character n-grams, misspellings, mispronunciations, frequency patterns of particular words, POS n-grams, content words, function words, Term Frequency-Inverse Document Frequency (TF-IDF), Continuous Bag of Words (CBOW), *etc.* [4,

Table 1. Number of training instances

Language	Instances
Malayalam (MA)	200
Bengali (BE)	202
Kannada (KA)	203
Telugu (TE)	210
Hindi (HI)	211
Tamil (TA)	207
Total	1233

6], whereas irrespective of the feature vector used, the most dominant choice of classifier is Support Vector Machine (SVM) [10].

1.1 Task Description

The corpus of INLI-2018 contains English comments/opinions of anonymous users that featured in regional newspapers, which was taken from Facebook. Users understood one of the following six native languages: Malayalam (MA), Bengali (BE), Kannada (KA), Telugu (TE), Hindi (HI), and Tamil (TA). The underlying assumption of this process is that only native speakers will read the native language newspapers. The distribution of the training instances with respect to the classes can be seen in Table 1. There are two test sets provided by the organizers (TestSet-1 and TestSet-2). TestSet-1 is the same test set that was used in INLI-2017, and TestSet-2 is a new test set given at INLI-2018 [3]. In total TestSet-1 and TestSet-2 has 783 and 1185 instances respectively. Detailed description about the training set and the TestSet-1 can be found in [2].

1.2 Summary of our approach

We have applied a supervised machine learning approach to tackle the task given at INLI-2018. The summary of our approach are as follows:

- Data preparation and preprocessing.
- Extract TF-IDF feature vectors.
- Build pairwise classifier models with linear-SVM as described in [7].
- Predict class label for the test instances

The remainder of this document is as follows. In Section 2, we will discuss our methodology. The achieved identification accuracy will be presented in Section 3 and we present concluding remarks in Section 4.

2 Methodology

We followed a multi-class classification approach as described in [7] for this task. The detailed description of our approach has been given below.

2.1 Data preprocessing

The given corpus collected from social media includes a significant number of non-ASCII characters like emojis. In general, user-generated social media texts are very noisy and contains irrelevant textual information for the classification process. Therefore, we remove such non-ASCII characters before feature extraction process. We have also replaced multiple occurrences of some characters like "....." or "sorryyyyyyy" with "." or "sorry".

2.2 Feature extraction

TF-IDF is a well-known weighting algorithm that measures the importance of a word in a document, given a collection of documents. The rationale behind this algorithm is that, if a word appears in a document very frequently, then it should be a significant word for that document and should be given a higher weight. However if that word appears in many other documents, it is not a unique identifier and therefore it should be given lower weight. We have used this feature in our approach.

TF-IDF is a product of two different measures, Term Frequency (TF) and Inverse Document Frequency (IDF): $tf-idf(t,d,D) = tf(t,d) \times idf(t,D)$, where t denotes the words/terms; d denotes a given document; $D = d_1, d_2, \ldots d_n$ denotes the collection of the documents. The first part of this formula $tf(t,d)$ equates the number of times each word appeared in each document and it excludes stop words such as "a", "the" *etc.* . The second part of the formula is $idf(t,D) = log\frac{|D|}{1+|\{d \in D : t \in d\}|}$. Note that frequency of a term in the document does not affect IDF, and 1 is added to the denominator to avoid division by zero.

2.3 Classifier models and prediction

In the first run, we have used one-vs-rest classifier settings. In the second run we have randomly arranged the set of classes into pairs and for each pair (*i.e.* i and j) based on the classifier's probability, we determine if the data fits better into class i or class j, and then proceed to the next round of the scheme [7, 8]. In the third run, for each class i, we randomly choose k other classes and determine the mean score for class i when comparing the test data in k pairwise comparisons with the randomly chosen classes. The class having the highest total score is selected as the identified class [7, 8]. We would like to mention that in all of the above approaches we have used linear-SVM with l2 regularization as a choice of the classifier.

3 Result

We report our achieved results using three evaluation metrics and it's overall accuracy. The three measures are: $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$ and $F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$ where TP is True Positive, FP is False

Table 2. Achieved results.

Run	Class	TestSet-1			TestSet-2		
		Precision (in %)	Recall (in %)	F-measure (in %)	Precision (in %)	Recall (in %)	F-measure (in %)
Run-1	BE	60.9	80	69.2	39	37.7	38.3
	HI	50	3.2	6	9.3	2.9	4.4
	KA	31.4	51.4	39	31.9	38.4	34.8
	MA	30.8	70.7	42.9	28.6	41.5	33.9
	TA	40.5	45	42.7	26.8	40.7	32.3
	TE	32.1	32.1	32.1	42.8	23.6	30.4
	Overall Accuracy	42.1%			31.8%		
Run-2	BE	56	75.1	64.2	35.7	38.6	37.1
	HI	53.8	2.8	5.3	8.6	2.2	3.5
	KA	29.3	45.9	35.8	32.7	36.8	34.7
	MA	28.3	71.7	40.6	26.9	43	33.1
	TA	39.6	42	40.8	25.2	37.1	30.1
	TE	35.8	29.6	32.4	43.7	20.8	28.2
	Overall Accuracy	39.8%			30.8%		
Run-3	BE	56.4	78.9	65.8	35.2	39.6	37.3
	HI	53.8	2.8	5.3	8.6	2.2	3.5
	KA	27.5	48.6	35.1	33	40	36.2
	MA	30.4	71.7	42.7	27.7	43	33.7
	TA	39.1	36	37.5	27	36.4	31
	TE	35.2	30.9	32.9	44.7	20.4	28
	Overall Accuracy	40.4%			31.5%		

Positive, and FN is false negative predicted values. Table 2 shows the achieved results obtained during different runs of the two test sets. We can clearly observe very poor performance for HI and good performance for BE irrespective of the test set. We also observe that the overall performance for TestSet-2 is poor when compared to the TestSet-1.

We observe that Run-1 performed better than other runs. We believe that Run-2 and Run-3 techniques could perform well when we have many more classes present for identification.

4 Conclusion

We have discussed our methodology used to solve the task given at INLI-2018 and have derived some insights from the achieved results. The achieved results (*i.e.* 42.1% for test TestSet-1 and 31.8% for test TestSet-2) are moderate when compared to the other techniques used in this task. We believe that improving the feature set could improve the results.

References

1. Abbasi, A., Chen, H.: Applying authorship analysis to extremist-group web forum messages. IEEE Intelligent Systems **20**(5), 67–75 (2005)
2. 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. In: CEUR Workshop Proceedings. vol. 2036, pp. 99–105 (2017)
3. Anand-Kumar, M., Barathi-Ganesh, H.B., Soman, K.P.: Overview of the inli@fire-2018 track on indian native language identification. In: CEUR Workshop Proceedings (2018)
4. Gebre, B.G., Zampieri, M., Wittenburg, P., Heskes, T.: Improving native language identification with tf-idf weighting. In: BEA@NAACL-HLT (2013)
5. Malmasi, S., Evanini, K., Cahill, A., Tetreault, J., Pugh, R., Hamill, C., Napolitano, D., Qian, Y.: A report on the 2017 native language identification shared task. In: 12th Workshop on Building Educational Applications Using NLP (2017)
6. Mechti, S., Abbassi, A., Belguith, L.H., Faiz, R.: An empirical method using features combination for arabic native language identification. In: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA). pp. 1–5 (2016)
7. Mondal, S., Bours, P.: Person identification by keystroke dynamics using pairwise user coupling. IEEE Transactions on Information Forensics and Security **12**(6), 1319–1329 (2017)
8. Mondal, S., Bours, P.: A continuous combination of security & forensics for mobile devices. Journal of Information Security and Applications **40**, 63 – 77 (2018)
9. Perkins, R.: New Threats and Countermeasures in Digital Crime and Cyber Terrorism, chap. Native Language Identification (NLID) for Forensic Authorship Analysis of Weblogs, pp. 213–234. IGI Global (2018)
10. Stehwien, S., Pad, S.: Generalization in native language identification – learners versus scientists. In: Proceedings of CLiC-IT. pp. 264–268. Trento, Italy (2015)
11. Tetreault, J., Blanchard, D., Cahill, A.: A report on the first native language identification shared task. In: In Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications (2013)