Bits_Pilani@INLI-FIRE-2017:Indian Native Language Identification using Deep Learning

Rupal Bhargava*
Birla Institute of Technology and Science, Pilani Campus
Pilani, India

Jaspreet Singh
Birla Institute of Technology and Science, Pilani Campus
Pilani, India

Shivangi Arora
Birla Institute of Technology and Science, Pilani Campus
Pilani, India

Yashvardhan Sharma
Birla Institute of Technology and Science, Pilani Campus
Pilani, India

ABSTRACT

The task of Native Language Identification involves identifying the prior or first learnt language of a user based on his writing technique and/or analysis of speech and phonetics in second language. There is a surplus of such data present on social media sites and organised dataset from bodies like Educational Testing Service(ETS), which can be exploited to develop language learning systems and forensic linguistics. In this paper we propose a deep neural network for this task using hierarchical paragraph encoder with attention mechanism to identify relevant features over tendencies and errors a user makes with second language for the INLI task in FIRE 2017. The task involves six Indian languages as prior/native set and english as the second language which has been collected from user's social media account.

CCS CONCEPTS

• **Computing methodologies → Natural language processing**; *Machine learning*; • **Information systems → Information systems applications**;

KEYWORDS

Native Language Identification, Natural Language Processing, Deep Learning, Neural Network, Machine Learning

1 INTRODUCTION

Native Language Identification (NLI) is the task of identifying the native language of a user based on his usage of second language with the help of a computer program.The task is usually modelled as a classification problem where a machine learning algorithm is trained in a supervised fashion, which is then used for predicting the native language of the user text.

NLI works by underpinning the fact that users′ linguistic background will lead them to use particular language phrases/styles more oftenly in their newly acquired languages. Despite the increasing research in this field there is lack of NLI datasets covering the wide span of languages and are still pretty small in size.

NLI is a non-trivial and challenging problem with an assumption that the native or first language influences Second Language Acquisition (SLA) [7]. If machines could learn tendencies and mistakes that language learners make, then it would help in development of education systems for learning new languages and acquisition. It would also help educators to develop techniques for helping learn difficult aspects of second language based on their first language and its language transfer pattern [2, 5].NLI can be closely related to the task of authorship profiling, which aims to extract information like age, gender and native origin of the author solely from text which is useful for forensic linguistics. NLI can be used to improve the performance of automatic speech recognition (ASR) for non-native speakers using speech and phonetic features for the task.

2 RELATED WORK

NLI as an artificial intelligence challenge is gaining popularity, which can be seen it being part of several shared tasks in various events in recent years[13, 16, 18]. Usually, models will try to extract patterns that a speakers with different native language will have in terms of different topic biases, misspellings, mispronunciations or usage frequency of particular words. Also some languages have specific linguistic styles, like Japanese is much more formal in nature. Malmasi [12] have extensively tested series of linear classifiers, and observed that state of the art results are achived by ensemble model. The features they have used are simple unigrams, bigrams, and character n-grams further including function word, POS-tagged n-grams and sentence dependencies for improving the results. Usually character level features generally outperform word level features for NLI. Stehwein and Pado [17] also analyze the performance of SVM's on this task, and use their results to identify key features of the datasets. SVM's tend to outperform neural networks when examined for performance on this task with a similar dataset by Malmasi et al [11]. Deep Neural networks have not been used much for NLI task, even in 2013 shared task there was no deep neural network submission[18]. Previous approaches were dependent upon features like grammatical structure of the language, string kernels[4], syntactic features [3]. In forthcoming sections the design and performance of our model is described.

3 DATA ANALYSIS

Dataset provided by task organizers[10] contains information collected from English speakers of six different native Indian languages. It includes 1233 written text by the different speakers on social media websites. All the data was present in romanised script. The distribution of class and training instances can be seen in Table 1.

4 PROPOSED TECHNIQUE

We have tried to model the task as a text classification problem and have tried solving it using hierarchical encoder [19], so the task

[*] Corresponding author e-mail address: rupal.bhargava@pilani.bits-pilani.ac.in

Table 1: Dataset

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

we had was to pre-process the text for passing it to network and generate word embeddings and design the neural networks model.

4.1 Pre-Processing

The data is tokenized, and capitalizations are removed. The english stop words have not been removed and as well as the punctuation marks are also retained, as they might be useful information to classify into the native language. Function words such as ‘which’, ‘the’, ‘at’, have been useful to distinguish native language[6]. Fixed length sentence runs are formed by delimiting with full-stop, comma and semi-colon which were padded by zeros to keep 128 as the fixed length input to network.

4.2 Word Embeddings

In each of three different runs we have used a different approach for generating the word vectors. The combined testing and training data has around 23,000 unique tokens in roman script but contains slang and transliterated native language words. Below are the different inputs methods of our embeddings we tested:

4.2.1 Pre-trained vectors. : We used google news embeddings which are produced by word2vec [14] model having a vocabulary of 3 million words and phrases and dimensionality of 300 for Run 1 which were further trained by applying online learning jointly over the training and testing corpus for Run 2, which gives the embedding additional context over the text that has to be dealt with. We think that most pretrained word vectors will fail to cover big parts of our vocabulary and even online learning is not enough for capturing context with such small corpus.

4.2.2 Random initialized word vectors. : Due to the above pointed short comings in pre-trained vectors, we also used randomly initialized vectors of dimension 300 only, as embeddings for Run 3, and trained these embeddings through backpropagation during model fitting, while this has ability to build embeddings even more effectively except model requires more data to train and has risk of over-fitting.

4.3 Classification

The most intuitive design for a text classifying neural network is a recurrent architecture due to their retention of longer term dependencies, and bi-directional one can also capture context in reverse order. We have used GRU[1] cells instead of LSTM as the give equal performance with lesser training. The first two runs had network of similar depth the bi-directional layer has 256 GRU units for sentence encoding and 128 GRU units for paragraph encoding. But for randomly initialized embedding which had to be trained the model was kept shallow having half the number of GRU cells for both the encoders. The both the encoders have attention layer added after recurrent units which help the model to weight words and sentences which effectively classify it. For paragraph encoder the final hidden state of the attention layer was fed to a fully connected softmax layer which returned probability distribution of six classes.

5 ALGORITHM

This project deals with classification of social media text to its correct native language of the user. Basic assumption is that a text has K sentences s_i and each sentence contains T_i words. w_{it} where $t \in [1, T]$ represents the words in the ith sentence. The model encodes the raw text into a vector representation, which is passed to a neural network to perform text classification. Below we have described building of text level vector from word vectors by using two levels of encoding [8, 15] represented in Figure 1[1].

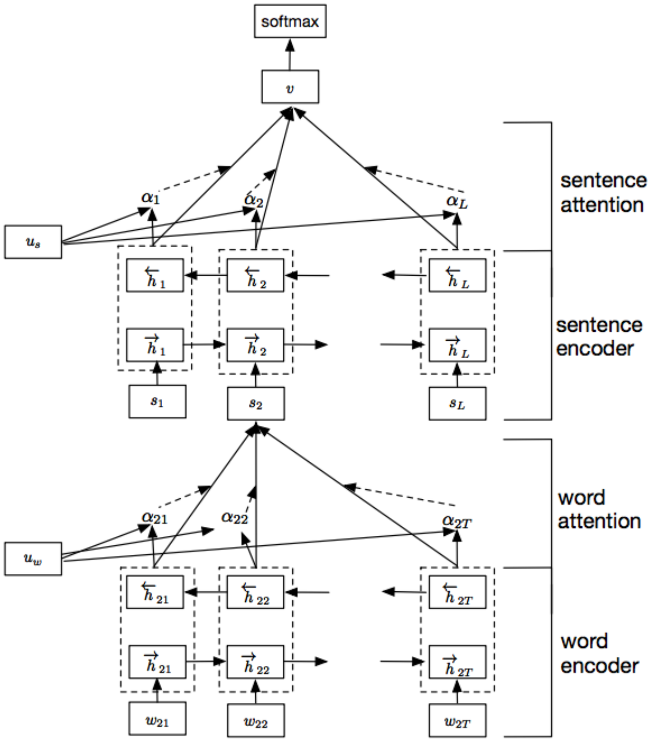

Figure 1: Hierarchical Attention

5.1 Word Level Layers

5.1.1 Word Encoder. It takes sentence as input and if a sentence has words w_{it} where $t \in [1, T]$, we first convert the words to vectors using the embedding matrix created above. We use a bidirectional Gated Recurrent Unit [1] to get representation of words as

[1] Image from Hierarchical Attention Networks for Document Classification Yang et al.(2016) under Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License

h_{it}, which contains the information of the whole sentence centred around w_{it} from both directions.

5.1.2 Word Attention. Next, the above computed hidden representation is subject to word attention layer[9], as some words are more important for representation of the sentence meaning. For that we pass the word annotation h_{it} through a single layer multilayer perceptron to get a hidden representation u_{it}, then the importance weight for a word is computed by similarity of u_{it} with a word level context vector u_w by a softmax function. After that, sentence vector s_i is computed as the weighted sum of the word annotations.

5.2 Sentence Level Layer

5.2.1 Sentence Encoder. The sentence vector s_i generated by word encoder is passed to the sentence encoder, a similar hidden representation h_{it} is returned by bidirectional Gated Recurrent Unit whose count of units was described in section 4.3, but this time it has paragraph level context.

5.2.2 Sentence Attention. As to weight sentences that are more relevant for classification, we use attention layer and introduce a sentence level context vector u_s and again use the softmax function for similarity calculations. The text vector v is weighted sum of encoded sentences. Further this vector v is passed to fully connected softmax layer to generate class probabilities.

6 EXPERIMENTS & RESULTS

We used an 80-20% split of the training data to validation split. All three model had a training accuracy of near 95% and had validation accuracies in the range of 60% - 70%, below is the confusion matrix of second run, the better of other two run over validation split.

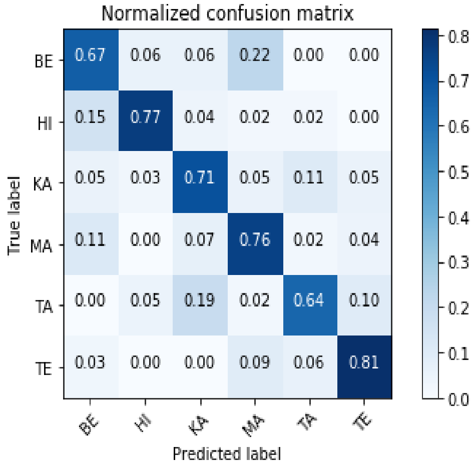

Figure 2: Confusion Matrix

6.1 Evaluation & Discussion

The test set contained 783 unlabelled files and the submission of labelled was evaluated by the task organizer, primarily on the basis of accuracy but precision, recall and F1 score were also recorded.

Table 2: Results

Runs	Lang	Precision	Recall	F1-Score	Accuracy
Run1	BE	39.70	15.70	22.50	26.90
	HI	24.00	19.50	21.50	
	KA	30.40	45.90	36.60	
	MA	20.00	28.30	23.40	
	TA	26.60	37.00	31.00	
	TE	28.80	44.40	35.00	
Run2	BE	56.30	38.40	45.70	28.00
	HI	23.90	6.80	10.60	
	KA	26.00	45.90	33.20	
	MA	15.50	31.50	20.80	
	TA	22.50	39.00	28.60	
	TE	30.50	35.80	33.00	
Run3	BE	39.40	23.20	29.30	26.70
	HI	19.00	8.80	12.00	
	KA	20.80	59.50	30.80	
	MA	34.30	39.10	36.50	
	TA	21.50	40.00	28.00	
	TE	43.60	29.60	35.30	

We submitted three runs whose training has been described at the beginning of the section, our highest performance was obtained by second run as depicted in Table 2.

As per the results published by the task organizer[10] the highest accuracy of each team is shown below(Figure 3). As it can be seen despite good training and validation accuracies, the model did not generalize sufficiently and possibly suffered from problems of low context embeddings and vocabulary shortage.

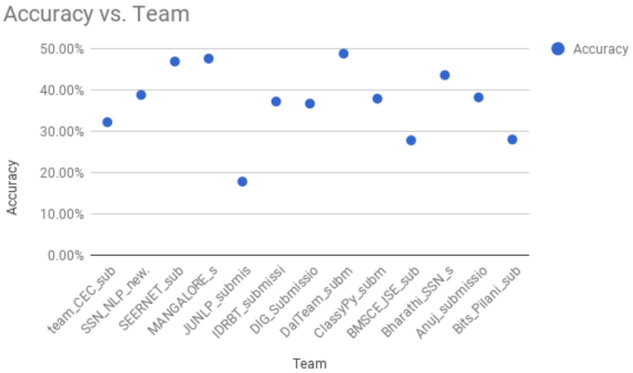

Figure 3: Team Accuracies

6.2 Error Analysis

The paper aims to develop a system which performs effective native language identification without the use of grammatical and structural features of languages. Although the model is good but fails to capture generalized features and performs poorly on the test set. The major issue is lack of ample data for effective training for a deep learning model. Another issue for a such word level model is large amount of slang and transliterated words present in data whose context is not very effectively captured by word embedding for such small sample of training corpus, moreover it suffers from vocabulary shortage over the test data which affects the classification performance.

7 CONCLUSION & FUTURE WORK

In this paper, we have outlined a native language identification approach for Indian languages based on an hierarchical deep neural network. We describe our system for the INLI Task in FIRE 2017, which involves a neural approach to this task. Although deep neural networks are able to learn features for this task, traditional methods still perform better with current datasets and models. In future we plan to continue to work on this problem and develop a hybrid system which combines traditional approaches of POS-tagged n-grams and sentence dependencies with deep learning models.

ACKNOWLEDGMENTS

We are thankful to the authorities of Department of Computer Science at Birla Institute of Technology and Science, Pilani, for providing infrastructure facilities for pursuing the work done for this research project.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Julian Brooke and Graeme Hirst. 2012. Measuring Interlanguage: Native Language Identification with L1-influence Metrics.. In *LREC*. 779–784.

[3] Serhiy Bykh and Detmar Meurers. 2014. Exploring Syntactic Features for Native Language Identification: A Variationist Perspective on Feature Encoding and Ensemble Optimization.. In *COLING*. 1962–1973.

[4] Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. 2016. String kernels for native language identification: insights from behind the curtains. *Computational Linguistics* (2016).

[5] Scott Jarvis and Scott A Crossley. 2012. *Approaching Language Transfer Through Text Classification: Explorations in the Detectionbased Approach*. Vol. 64. Multilingual Matters.

[6] Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005. Automatically determining an anonymous authorâĂŹs native language. *Intelligence and Security Informatics* (2005), 41–76.

[7] Robert Lado. 1957. Linguistics Across Cultures: Applied Linguistics for Language Teachers. (1957).

[8] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057* (2015).

[9] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[10] Anand Kumar M, Barathi Ganesh HB, Shivkaran S, Sonam 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] Shervin Malmasi et al. 2016. Native language identification: explorations and applications. (2016).

[12] Shervin Malmasi and Mark Dras. 2017. Native Language Identification using Stacked Generalization. *arXiv preprint arXiv:1703.06541* (2017).

[13] Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano, and Yao Qian. 2017. A Report on the 2017 Native Language Identification Shared Task. In *Proceedings of the 12th Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, Copenhagen, Denmark.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[15] Nikolaos Pappas and Andrei Popescu-Belis. 2017. Multilingual hierarchical attention networks for document classification. *arXiv preprint arXiv:1707.00896* (2017).

[16] Björn W Schuller, Stefan Steidl, Anton Batliner, Julia Hirschberg, Judee K Burgoon, Alice Baird, Aaron C Elkins, Yue Zhang, Eduardo Coutinho, and Keelan Evanini. 2016. The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity & Native Language.. In *INTERSPEECH*. 2001–2005.

[17] Sabrina Stehwien and Sebastian Padó. 2015. Generalization in Native Language Identification: Learners versus Scientists. *CLiC it* (2015), 264.

[18] Joel R Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A Report on the First Native Language Identification Shared Task.. In *BEA@ NAACL-HLT*. 48–57.

[19] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J Smola, and Eduard H Hovy. 2016. Hierarchical Attention Networks for Document Classification.. In *HLT-NAACL*. 1480–1489.