NLPRL@INLI-2018: Hybrid gated LSTM-CNN model for Indian native language identification

Rajesh Kumar Mundotiya[1], Manish Singh[2], and Anil Kumar Singh[1]

[1] Department of Computer Science and Engineering, IIT(BHU), Varanasi
{rajeshkm.rs.cse16, aksingh.cse}@iitbhu.ac.in
[2] Department of Linguistics, BHU, Varanasi
maneeshhsingh100@gmail.com

Abstract. Native language identification (NLI) focuses on determining the native language of the author based on the writing style in English. Indian native language identification is a challenging task based on users comments and posts on social media. To solve this problem, we present a hybrid gated LSTM-CNN model to solve this problem. The final vector of a sentence is generated at hybrid gate by joining the two distinct vector of a sentence. Gate seeks the optimum mixture of the LSTM and CNN level outputs. The input word for LSTM and CNN are projected into high-dimensional space by embedding technique. We obtained 88.50% accuracy during training on the provided social media dataset, and 17.10% is reported in the final testing done by Indian native language identification (INLI) workshop organizers.

Keywords: Bi-LSTM · CNN · Glove.

1 Introduction

Native Language Identification is a process to automatically identify the native language of an author by the writing or the speaking accent of his or her in another language that is acquired as a second language [1]. It can identify the writing structure based on the authors linguistic background. It can be used for several applications namely authorship profiling and identification, forensic analysis, second language identification and educational applications. English is one of the well known and commonly used languages among humans. In this shared task, the goal is to identify the Indian native language written on social media as post or comment in English. Indian native language includes Bengali, Hindi, Kannada, Malayalam, Tamil and Telugu. The assumption behind this dataset collection is that only native language speakers will read native language newspapers [1] [12].

We have tackled this problem by supervised learning as classification problem but the main challenges for this are insufficient dataset size. There are couple of datasets used in past research which are freely available. International Corpus of Learner English (ICLE)[3] corpus is one of the first appearing in the early studies.

[3] https://uclouvain.be/en/research-institutes/ilc/cecl/corpora.html

It was publicly used for prediction of native language of learner based on his/her writing style. It was released in 2002 and updated in 2009.

In the following sections; we mentions related work in Section 2, we describe the proposed model and training procedure in Section 3, we show the result and analysis in Section 4 and finally, draw conclusion in Section 6.

2 Related Work

Native language identification is a new and significant problem. Language learners are prone to make mistakes similarly if machines can learn the same tendencies of making mistakes then it may help in developing systems for educational domain. Several researchers worked on this problem and similar problem like second language acquisition. One of the earliest work, Tomokiyo and Jones (2001) tried to discriminate non-native statements from native statements, written in English by Nave Bayes [2].
Kochmar et al. (2011) has performed experiments on prediction of the native languages of Indo-European learners. He treated this problem as binary classification and use linear kernel SVM. The features used for prediction were n-grams and words. Also, The errors were tagged manually within the corpus [3]. Besides this, some other [4], [5] also used the SVM with different features.
In the recent past, word embedding and document embedding has gained much attention along with other features. Vector representations for documents were generated with distributed bag-of-words architectures using Doc2Vec tool. The authors developed a native language classifier using document and word embedding with an accuracy of 82% for essays and 42% on speech data [6]. Yang et al. (2016) have purposed hierarchical attention network for classification problem. They required vast corpus size attend the significant word and sentence by attention mechanisms [10]. Kim (2014) used the convolutional neural network and got a state-of-the-art accuracy, but it can hold contextual information till window-size. [11]
In 2017 another shared task on NLI was organized. The corpus included essays and transcripts of utterances. According to Malmasi et. al. (2017), the ensemble methods and meta-classifiers with syntactic or lexical features were the most effective systems [7].

3 Model Description

The model architecture is showed in figure 1.

Each document includes m sentences and each sentence within a document consists of n words. The word level input w_{i*n} projected into high-dimensional vector space with the help of pretrained glove English word embedding, $w_{i*n} \in \Re^k$, where k is word vector's dimension, i and n represents sentence and word respectively. The word level input is converted into sentence level input by using a

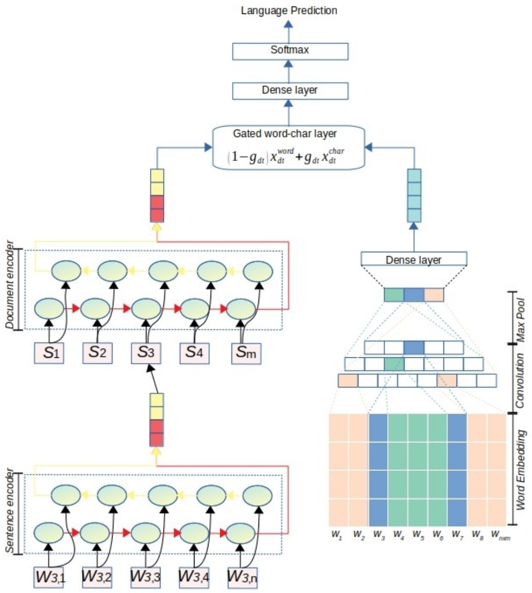

Fig. 1. Hybrid gated LSTM-CNN system architecture.

bidirectional LSTM. This is achieved by linearly combining the last hidden state of forward and backward LSTM.

$$x_{st}^{word} = W_s^f h_f^{st} + W_s^r h_r^{st} + b_s \tag{1}$$

Where, $s_t \in S$, S is the total sentences in a document and $h_{st}^f, h_{st}^r \in \Re^k$ are the forward and backward LSTM's end hidden states, respectively. $w_s^f, w_s^r \in \Re^{k*k}$ and $b_s \in \Re^k$ are trainable parameters and $x_{st}^{word} \in \Re^k$ is the final vector representation of the sentence s_t. Similarly, with the help of bidirectional LSTM, we will retrieve the single vector representation of the document. It will take vector representation of the sentences [10] as input.

$$x_{dt}^{word} = W_d^f h_f^{dt} + W_d^r h_r^{dt} + b_d \tag{2}$$

Where $d_t \in D$, D is the number of documents which is composed of the combination of the last hidden state of the forward and backward LSTM, represented by $h_{dt}^f, h_{dt}^r \in \Re^k$ respectively. $w_d^f, w_d^r \in \Re^{k*k}$ and $b_d \in \Re^k$ are trainable parameters and $x_{dt}^{word} \in \Re^k$ is the final vector representation of the sentence d_t.
Convolutional neural network [11] helps to extract features by applying convolving filters on input word vector. Let $w_t \in \Re^k$, where k^{th} is the dimensional vector like to the i^{th} word in the document. A document length of length $n*m$ is represented as:

$$w_{1:n*m} = w_1 \oplus w_2 \oplus w_3 \oplus w_4 \oplus, ..., \oplus w_{n*m} \tag{3}$$

A convolution operation involves a filter and the size of this filter size shows h-gram of words. The filter moves over the entire document $\{w_{1:h}, w_{2:h+1}, ...$ $, w_{n*m-h+1:n*m}\}$ and generate features $\{c_1, c_2, ..., c_{n*m-h+1}\}$. These generated features produce a feature map c.

$$c = [c_1, c_2, ..., c_{n*m-h+1}] \tag{4}$$

Now we want to extract most important feature from feature map, so we apply max-over-pooling on this feature map that gives maximum value $\hat{c}= max\{c\}$ as the feature crossponding to this particular filter. This process extracts one feature from one filter. This model uses multiple filters with different h-gram to extract multiple features. These different extracted features concatenated and flatten before passing in the dense layer. We have D documents, for each document the entire process is repeated. The final vector representation of the sentence d_t is $x_{dt}^{char} \in \Re^k$ where $d_t \in D$.
The assumption is that, bidirectional LSTM can capture the entire document features regarding language model into x_{dt}^{word} and n-gram features for convolutional neural network into x_{dt}^{char}. If both combine, then it may help to distinguish the Indian languages. The gated word-char layer combines these two types of vector generation from bidirectional LSTM and convolutional neural network. The process of mixing generated vectors x_{dt}^{word} and x_{dt}^{char} is done by gate g_{dt}, where g_{dt} is also a dense layer with sigmoid activation function.

$$g_{dt} = \sigma(v_g^T x_{dt}^{word} + b_g) \tag{5}$$

$$x_{dt} = (1 - g_{dt}) x_{dt}^{word} + g_{dt} x_{dt}^{char} \tag{6}$$

Where $v_g \in \Re_d$ and $b_g \in \Re$ are, weight vector and bias respectively which are trainable parameters. Miyamoto and Cho (2016) applied this approach at word level on language model to handle out-of-vocabulary and unusual words [8]. Hashimoto and Tsuruoka (2016) has also applied a very similar approach on compositional and non-compositional phrase embedding and achieved state-of-the-art score on verb disambiguation and compositionality detection tasks [9]. The features from the gated word-char layer pass to the dense layer with softmax. The output is the probability distribution over languages.

$$p = softmax(v_s^T x_{dt} + b_s) \tag{7}$$

Where v_s and b_s are, weight vector and bias respectively which are trainable parameters.

4 Experiment

4.1 Dataset

We have evaluated our model on the dataset provided by the task organizers. The dataset contains English comments of Facebook users. English comments are written in native language by users whose native language includes Bengali (BE), Hindi (HI), Kannada (KA), Malayalam (MA), Tamil (TA) and Telugu (TE). The statistics of the train dataset are summarized in Table 1. We divided the training dataset into training and validation data in the ratio of 90:10.

Table 1. Training data statistics

Language	# of Documents in train dataset
Bengali	202
Hindi	211
Kannada	203
Malayalam	200
Tamil	207
Telugu	210

4.2 Model configuration and training

The given dataset is preprocessed to remove multiple occurrences of punctuation symbols which occurred in between the sentences or at the end of the sentences. Periods, question marks and exclamation marks are some of the punctuation symbols that occurred in the available dataset. Punctuation symbols present at the end of the sentence defines the sentence boundary hence such kinds of repeated symbols are replaced by single symbol. There were some occurrences of emojis which are also removed. After preprocessing the obtained list of words is lemmatized[4] to obtain the root word using the grammatical rules of the concerned language. There is variation in the occurrence of number of sentences and number of words in a sentence for different language document. To overcome this issue, we perform padding.

Each word is represented in 100 dimensions vector using pre-trained glove embedding. Glove embedding [5] is trained on 6 billion tokens of the combination of Gigaword5 + Wikipedia2014 dumps. Some words are not present in glove embeddings, which are represented by the glove vector dimension of random uniform distribution range from -1 to +1. We have set 20 as maximum document length and 300 as maximum sentence length. Number of hidden units is fixed as 100 for each directional layer at sentence level and document level. The convolutional neural network uses three convolutional layer whose filter size is 2, 3 and 5 respectively however stride size is fixed as 1 for all convolutional layer. A dense layer of the convolutional neural network has 100 hidden units that represents a document. We have 6 prediction classes (HI, BE, KA, MA, TA, TE), so number of hidden units in final prediction layer will be based on number of prediction classes.

[4] $https : //www.nltk.org/_modules/nltk/stem/wordnet.html$

[5] https://nlp.stanford.edu/projects/glove/

The entire network is trained by Adam optimizer with epoch and mini-batch size of 15 and 10 respectively. Model is implemented on GeForce 840 GPU using keras [6] neural network library.

5 Result and Analysis

The task organizers evaluated the model on different metrics such as precision, recall and F-score for each language with overall accuracy. Two different test sets are provided for evaluating the model. The number of document in test set1 and test set2 are 783 and 1185 respectively. The obtained result is given in Table 2. We obtain the overall accuracy of 15.3% for test set1 and 17.1% for test set2 Indian native language identification. Our model retrieved more relevant documents for Tamil language as compared to other languages during testing phase. For Hindi and Tamil language, our model achieves highest F1-score for Testset1 data.

We obtained accuracy of 88.5% on training dataset however there is significant drop in the result during the testing phase which indicates over-fitting as a probable reason. There is significant difference in the number of documents shared for the training and testing phase. Considering the size of trainable data used during the training phase, our model was unable to predict more generalized features due to this large number of random word embeddings were formed which inturn reflects poor results during the testing phase.

Table 2. Performance analysis on Testset1 and Testset2

	Testset1			Testset2		
Class	Prec.(%)	Rec.(%)	F1(%)	Prec.(%)	Rec.(%)	F1(%)
BE	0.00	0.00	0.00	47.60	4.80	8.80
HI	30.60	16.30	21.30	12.70	18.80	15.20
KA	10.40	21.60	14.00	18.00	14.40	16.00
MA	2.70	2.20	2.40	15.90	13.00	14.30
TA	14.50	40.00	21.30	12.00	33.60	17.70
TE	15.00	25.90	19.00	28.40	23.20	25.60
Overall	15.30			17.10		

6 Conclusion

The purposed hybrid gated LSTM-CNN model for identifying the Indian native languages, namely Bengali, Hindi, Kannada, Malayalam, Tamil and Telugu, posted on social media in English by the native speaker of those languages.

[6] https://keras.io/

Each document is represented in a vector by combining the output obtained using bidirectional-LSTM and convolutional neural network. We obtained better accuracy in test set2 data as compared to the accuracy obtained in the test set1 data. The proposed system can be improved by applying hybrid gated LSTM-CNN model at word level instead of document level with dropouts.

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. Tomokiyo, L. M., & Jones, R. (2001, June). You're not from'round here, are you?: naive Bayes detection of non-native utterance text. In Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies (pp. 1-8). Association for Computational Linguistics.
3. Kochmar, E. (2011). Identification of a writers native language by error analysis (Doctoral dissertation, Masters thesis, University of Cambridge).
4. Mechti, S., Abbassi, A., Belguith, L. H., & Faiz, R. (2016, November). An empirical method using features combination for Arabic native language identification. In Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of (pp. 1-5). IEEE.
5. Kosmajac, D., & Keselj, V. (2017). DalTeam@ INLI-FIRE-2017: Native Language Identification using SVM with SGD Training. In FIRE (Working Notes) (pp. 118-122).
6. Vajjala, S., & Banerjee, S. (2017). A study of N-gram and Embedding Representations for Native Language Identification. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications (pp. 240-248).
7. Malmasi, S., Evanini, K., Cahill, A., Tetreault, J., Pugh, R., Hamill, C., ... & Qian, Y. (2017). A report on the 2017 native language identification shared task. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications (pp. 62-75).
8. Miyamoto, Y., & Cho, K. (2016). Gated word-character recurrent language model. arXiv preprint arXiv:1606.01700.
9. Hashimoto, K., & Tsuruoka, Y. (2016). Adaptive joint learning of compositional and non-compositional phrase embeddings. arXiv preprint arXiv:1603.06067.
10. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 1480-1489).
11. Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.
12. Anand Kumar M, Barathi Ganesh B and Soman K P. "Overview of the INLI@FIRE-2018 Track on Indian Native Language Identification. In: In workshop proceedings of FIRE 2018, FIRE-2018, Gandhinagar, India, December 6-9, CEUR Workshop Proceedings.