BERT-based Language Identification in Code-Mix Kannada-English Text at the CoLI-Kanglish Shared Task@ICON 2022

Pritam Deka
Queen's University Belfast
UK
pdeka01@qub.ac.uk

Nayan Jyoti Kalita
Gauhati University
Assam, India
nayan.jk.123@gmail.com

Shikhar Kumar Sarma
Gauhati University
Assam, India
sks@gauhati.ac.in

Abstract

Language identification has recently gained research interest in code-mixed languages due to the extensive use of social media among people. People who speak multiple languages tend to use code-mixed languages when communicating with each other. It has become necessary to identify the languages in such code-mixed environment to detect hate speeches, fake news, misinformation or disinformation and for tasks such as sentiment analysis. In this work, we have proposed a BERT-based approach for language identification in the CoLI-Kanglish shared task at ICON 2022. Our approach achieved 86% weighted average F-1 score and a macro average F-1 score of 57% in the test set.

1 Introduction

Social media plays a big role in today's life. With the deep penetration of the internet among the masses, people use social media in all directions. In a region where people use different languages, mixing words or sentences from more than one language is very common. This also happens on social media where people exchange their views using code-mixed languages, most of the time in a common script like Roman. (Bokamba, 1989) defined code-mixing as the blending of words or sentences between two distinct languages within a single speech occurrence. It has emerged as a separate language phenomenon in a multilingual culture as a result of the increased usage of social media (Das and Gambäck, 2015).

Although the problem of language identification is very old, major research has been done around the world on identifying languages in code-mixed environments (Al-Badrashiny and Diab, 2016; Shirvani et al., 2016; Volk and Clematide, 2014; Carpuat, 2014; Xia, 2016; Piergallini et al., 2016; Samih et al., 2016; Jaech et al., 2016). However, in a code-mixed scenario, there are relatively few studies that have attempted to find regional languages from India. In this paper, we have explored the use of state-of-the-art NLP and deep learning techniques to identify language in the CoLI-Kenglish dataset (Hosahalli Lakshmaiah et al., 2022) for the shared task CoLI-Kanglish (Balouchzahi et al., 2022). We also share our code used for the experiments on GitHub[1].

As a result of recent developments in NLP, a large number of language models built on the transformer paradigm have emerged (Vaswani et al., 2017). In terms of several NLP tasks, such as text categorization, natural language inference, question answering, and textual similarity, one such model, called BERT, has produced state-of-the-art results (Devlin et al., 2018). These models can be used for a variety of downstream tasks because they were trained on massive amounts of text data from sources like Wikipedia and BookCorpus. For our work, we have used BERT (Devlin et al., 2018) and deep neural networks for the Kannada-English language identification task. Our results evaluated on the test set show that using BERT can produce good results, which shows the potential of such models for future related work.

2 Related work

This section contains a brief discussion of some recent works on identifying languages in code-mixed language pairings for Indian languages.

(Chakravarthi et al., 2022) performed a sentiment analysis and offensive language identification on a data set collected from YouTube with approx 60,000 comments. They mainly focused on three Dravidian languages - Tamil, Kannada, and Malayalam. In the experiment, SVM, BERT (Devlin et al., 2018), DistilBERT (Sanh et al., 2019), CharacterBERT (Boukkouri et al., 2020), ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019),

[1] https://github.com/pritamdeka/CoLI-Kanglish

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 12 - 17
December 15-18, 2022

XLM (Lample and Conneau, 2019) and XLM-R are used. They found that classification algorithms performed better in sentiment analysis than offensive language detection.

A similar work was done by (Saumya et al., 2021) where the authors focused on offensive language detection from code-mixed Tamil-English, Malayalam-English pair and Malayalam language. In their experiment, as conventional learning models, they used SVM, Logistic Regression, Naive Bayes and Random Forest models. They also used BERT-base, BERT-multilingual and ULM-FiT (Howard and Ruder, 2018) as transfer models. They found that conventional learning models with character 1 to 6 gram TF-IDF features performed better in comparison to transfer and neural learning based models.

Similarly, (Balouchzahi et al., 2021) proposed two different models COOLI-Ensemble and COOLI-Keras to identify and classify code-mixed texts of three language pairs, namely, Kannada-English, Malayalam-English and Tamil-English into six predefined categories (5 categories in Malayalam-English language pair). The proposed models have been trained with features extracted from sentences such as character sequences combined with words. The authors found that the COOLI-Ensemble model performed the best among the proposed models.

Another work by (Thara and Poornachandran, 2021) focused on Malayalam-English code-mixed corpus at the word level using transfer models like CamemBERT (Martin et al., 2019), XLM-RoBERTa, ELECTRA (Clark et al., 2020) and DistilBERT. The results of this study showed that ELECTRA performed better than the other models.

Another recent study on language identification for Tamil code-mixed YouTube comments was conducted by (Vasantharajan and Thayasivam, 2022). The dataset was collected from YouTube posts and comments in a multilingual environment. CNN-BiLSTM, DistilBERT and XLM-R models gave similar but poor results on this dataset, and ULM-FiT attained a better performance over the other models due to its superior fine-tuning methods. They proposed a selective translation and transliteration for the code-mixed corpus. They also showed the advantage of using transformer based models on low resource languages.

3 Approach

We first describe the specifics of the dataset that we use in this section. After that we will discuss the approach that we used using BERT. We also compare the results among various BERT-based models along with traditional machine learning approaches.

3.1 Dataset details

The CoLI-Kenglish dataset(Hosahalli Lakshmaiah et al., 2022) consists of words written in Roman script that are both English and Kannada. These words are categorized into six main groups: “Kannada", “English", “Mixed-language", “Name", “Location" and “Other". Details of the dataset are shown in Table 1 and the statistics of the train set are shown in Table 2, both of which have been taken from the official shared task website[2].

Category	Tag	Description	Sample
Kannada	kn	Kannada words written in Roman script	kopista, baruthe. barbeku
English	en	Pure English words	small, need, take, important
Mixed-Language	kn-en	Combinations of Kannada and English words in Roman script	coolagiru, leaderge, homealli
Name	name	Words indicating name of a person (including Indian names)	Madhuswamy, Hemavati, Swamy
Location	location	Words indicating location	Karnataka, Bangalore
Other	other	Words not belonging to any of the categories and words of other languages	Znjdjfjbj- not a word, Kannada words in Kannada script, Hindi words in Devanagiri script, Hindi words in Roman script, Tamil words in Tamil script

Table 1: Dataset Details

Category	Tag	Count
Kannada	kn	6626
English	en	4469
Mixed-Language	kn-en	1379
Name	name	708
Location	location	102
Other	other	1663
Total		**14847**

Table 2: Statistics of the train set

3.2 BERT based neural network model

BERT (bidirectional encoder representations for transformers) (Devlin et al., 2018) is a transformer (Vaswani et al., 2017) language model and due to the state-of-the-art results in several NLP tasks, it caused a stir when it was released. To calculate

[2] https://sites.google.com/view/kanglishicon2022/dataset?authuser=0

word embeddings, BERT can be employed. Unsupervised pre-training of BERT has been done on BookCorpus and Wikipedia. It excels at producing semantically rich word vectors or embeddings that are heavily based on context. Due to the context of the words, BERT will produce entirely different word embeddings for the words “apple" in the sentences “I ate an apple" and “Apple acquired a startup". Older systems like word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) were less effective since the word embeddings did not adapt to the context of the nearby vector.

Our method involves the usage of a BERT-based word vector representation to represent the tokens found in the corpus and then using these representations as neural network training features. BERT is being used for this code mix corpus because of its capacity to learn contexts that can be used for language identification tasks. We describe the details of the experiment in the next section.

4 Experiment Details

For the BERT experiment purposes, we have used different BERT base models from HuggingFace[3]. We used the Tensorflow[4] framework for our experiments. We report the results of our experiments on the annotated test set of the dataset. For defining our neural network model, we have used three dense layers on top of the BERT embedding layer containing 128, 64 and 32 neurons, respectively, with *relu* activation function with a dropout rate of 0.2 at each layer. The final dense classification layer contains 6 neurons with a *softmax* activation function. The BERT layer consists of the word embeddings from the BERT-base model along with the input word ids and the masked sequence of the words. During the neural network model training we have used a learning rate of 2e-5 which is taken from the original BERT paper (Devlin et al., 2018). We used a maximum sequence length of 15, epsilon=1e-08, decay=0.01 and a batch size of 128 is used for the training over 20 epochs. We keep the same experimental settings for all the models. For optimization, we have used the Adam optimizer (Kingma and Ba, 2014) with a *categorical cross entropy* loss function

$$Loss, \delta = -\frac{1}{N}\sum_{i=1}^{N} \log p_m \Big[x_i \in A_{x_i}\Big] \quad (1)$$

where each x_i belongs to exactly one class, C_{x_i} and $p_m\Big[x_i \in A_{x_i}\Big]$ is the probability predicted by the model.

We calculated the weighted as well as macro average precision, recall and f-1 score on the test set for all experiments. The results are shown in Table 3. We also compared the results of traditional machine learning algortihms such as Logistic Regression, Multinomial Naïve Bayes, Random Forest and SVM shown in Table 4. The code for reproducing our results is available in GitHub[5].

5 Results and discussion

From the Table 3, we can see that BERT-base-uncased has the highest macro average F-1 score among all the other models. For comparison we have experimented with various models including DistilBERT (Sanh et al., 2019), RoBERTa (Liu et al., 2019), Deberta (He et al., 2020) and ELECTRA (Clark et al., 2020). It can be seen that DistilBERT, albeit having a smaller size, has a performance comparable to that of the BERT model. This is useful when there is less computation power and there should not be much decrease in performance of the model.

Model	Macro avg			Weighted Avg		
	P	R	F-1	P	R	F-1
BERT-base-uncased	0.57	0.58	0.57	0.87	0.86	0.86
DistilBERT-base-uncased	0.57	0.56	0.56	0.86	0.86	0.86
RoBERTa-base	0.56	0.50	0.52	0.85	0.85	0.84
Deberta-v2-base	0.54	0.50	0.51	0.84	0.84	0.83
ELECTRA-base-discriminator	0.56	0.51	0.50	0.85	0.83	0.82

Table 3: Comparison of transformer models

Among the traditional machine learning algorithms, SVM and Logistic Regression has similar macro F-1 scores which can be seen from Table 4. However, all of these algorithms perform poorly in comparison to the transformer models. This shows that learning the context behind words can lead to better results for the language identification task in a code-mixed language environment.

From the results we can see that using BERT, identification of languages in a code mix Kannada-English text corpus can be achieved with better results than traditional machine learning algorithms.

[3]https://huggingface.co/
[4]https://www.tensorflow.org/
[5]https://github.com/pritamdeka/CoLI-Kanglish

Since BERT can learn word contexts, our objective for adopting it is validated. As a result, it performs better when it comes to detecting languages with more precision and recall.

Machine Learning Algorithm	Macro avg			Weighted Avg		
	P	R	F-1	P	R	F-1
Multinomial Naïve Bayes	0.24	0.17	0.12	0.62	0.49	0.34
SVM	0.80	0.22	0.20	0.73	0.50	0.35
Logistic Regression	0.80	0.22	0.20	0.73	0.50	0.35
Random Forest	0.08	0.17	0.11	0.23	0.48	0.31

Table 4: Comparison of machine learning algorithms

We have also compared our work with the top ranked teams for the CoLI-Kanglish shared task. The results are shown in Tables 5 and 6. We can see that for the weighted average scores, our method has the same F-1 score as the top ranked team which is 86%. However, for the macro F-1 score, our method is lower than the rest of the teams with 57%.

Teams	Precision	Recall	F-1 Score
tiya1012	0.87	0.85	**0.86**
Abyssinia	0.85	0.84	0.84
Habesha	0.85	0.83	0.84
Lidoma	0.83	0.83	0.83
PDNJK (Ours)	0.86	0.85	**0.86**

Table 5: Comparison of weighted average scores with top ranked teams for the shared task

Teams	Precision	Recall	F-1 Score
tiya1012	0.67	0.61	**0.62**
Abyssinia	0.62	0.62	0.61
Habesha	0.66	0.60	0.61
Lidoma	0.64	0.56	0.58
PDNJK (Ours)	0.58	0.58	0.57

Table 6: Comparison of macro average scores with top ranked teams for the shared task

6 Ablation Study

We also performed a few ablation studies where we dropped a few of the category tags. From the Table 2 we can see that the tags “location" and “name" have less examples than the other categories. For our ablation studies, we first dropped only the “location" tag and performed the experiment with the BERT-base-uncased model. We then dropped only the “name" tag and performed the same set of experiment. We then dropped both tags and performed the experiment. The results of these studies are shown in Table 7.

Ablation Study Setting	Macro avg			Weighted Avg		
	P	R	F-1	P	R	F-1
Without “location" tag	0.65	0.64	0.64	0.85	0.87	0.86
Without “name" tag	0.74	0.59	0.57	0.91	0.88	0.89
Without “name" and “location" tags	0.70	0.73	0.71	0.91	0.90	0.90

Table 7: Ablation Study Results

We can see that dropping the “location" tag, we get an increased macro average F-1 score. However, the weighted average F-1 score remains the same. However, dropping only the “name" tag does not affect the macro average F-1 score. This shows that due to the less number of examples for the “location" tag, removing that tag increases the F-1 score. When we remove both tags, there is a significant increase in the F-1 scores. This shows that a smaller number of examples for “name" and “location" tags leads to poor model training. Therefore, having a higher number of examples for both tags may lead to increased training performance.

7 Conclusion

There is a large research potential for automatic language detection in code mix text. To spot hate speech or the dissemination of false information in a multilingual culture where speakers converse in a variety of languages, language identification is important. In this paper, we have used a BERT-based approach to identify language in a Kannada-English code mix corpus. We have seen improvements over traditional machine learning algorithms when using these models, paving the way for further research in this direction using such models. We have also seen that availability of more data can lead to increase in efficiency of such models.

References

Mohamed Al-Badrashiny and Mona Diab. 2016. The george washington university system for the code-switching workshop shared task 2016. In *Proceedings of The Second Workshop on Computational Approaches to Code Switching*, pages 108–111.

Fazlourrahman Balouchzahi, BK Aparna, and HL Shashirekha. 2021. Mucs@ dravidianlangtech-eacl2021: Cooli-code-mixing offensive language identification. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 323–329.

Fazlourrahman Balouchzahi, Sabur Butt, Asha Hagde, Noman Ashraf, Shashirekha Hosahalli Lakshmaiah, Grigori Sidorov, and Alexander Gelbukh. 2022. Overview of CoLI-Kanglish: Word Level Language Identification in Code-mixed Kannada-English Texts at ICON 2022. In *19th International Conference on Natural Language Processing Proceedings*.

Eyamba G Bokamba. 1989. Are there syntactic constraints on code-mixing? *World Englishes*, 8(3):277–292.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Junichi Tsujii. 2020. Characterbert: Reconciling elmo and bert for word-level open-vocabulary representations from characters. *arXiv preprint arXiv:2010.10392*.

Marine Carpuat. 2014. Mixed language and code-switching in the canadian hansard. In *Proceedings of the first workshop on computational approaches to code switching*, pages 107–115.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P McCrae. 2022. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *Language Resources and Evaluation*, pages 1–42.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Amitava Das and Björn Gambäck. 2015. Code-mixing in social media text: the last language identification frontier?

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Shashirekha Hosahalli Lakshmaiah, Fazlourrahman Balouchzahi, Anusha Mudoor Devadas, and Grigori Sidorov. 2022. CoLI-Machine Learning Approaches for Code-mixed Language Identification at the Word Level in Kannada-English Texts. *acta polytechnica hungarica*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Aaron Jaech, George Mulcaire, Mari Ostendorf, and Noah A Smith. 2016. A neural model for language identification in code-switched tweets. In *Proceedings of The Second Workshop on Computational Approaches to Code Switching*, pages 60–64.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de La Clergerie, Djamé Seddah, and Benoît Sagot. 2019. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Mario Piergallini, Rouzbeh Shirvani, Gauri Shankar Gautam, and Mohamed Chouikha. 2016. Word-level language identification and predicting codeswitching points in swahili-english language data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 21–29.

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual code-switching identification via lstm recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sunil Saumya, Abhinav Kumar, and Jyoti Prakash Singh. 2021. Offensive language identification in dravidian code mixed social media text. In *Proceedings of the first workshop on speech and language technologies for Dravidian languages*, pages 36–45.

Rouzbeh Shirvani, Mario Piergallini, Gauri Shankar Gautam, and Mohamed Chouikha. 2016. The howard university system submission for the shared task in language identification in spanish-english codeswitching. In *Proceedings of the second workshop on computational approaches to code switching*, pages 116–120.

S Thara and Prabaharan Poornachandran. 2021. Transformer based language identification for malayalam-english code-mixed text. *IEEE Access*, 9:118837–118850.

Charangan Vasantharajan and Uthayasanker Thayasivam. 2022. Towards offensive language identification for tamil code-mixed youtube comments and posts. *SN Computer Science*, 3(1):1–13.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Martin Volk and Simon Clematide. 2014. Detecting code-switching in a multilingual alpine heritage corpus. Association for Computational Linguistics.

Meng Xuan Xia. 2016. Codeswitching language identification using subword information enriched word vectors. In *Proceedings of the second workshop on computational approaches to code switching*, pages 132–136.