ICON 2022

Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts

Proceedings of the Shared Task

December 15-18, 2022

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-959429-38-8

Introduction

South Asia is the world's most linguistically diverse region, with over 650 languages. India, in particular, is a multilingual country with a rich language heritage that includes the Dravidian language Kannada. Kannada is the official and administrative language of the state of Karnataka, and has over 40 million native speakers. Many people in this region are comfortable using both English and their native language in daily communication. On social media platforms, multilingual speakers often use code-mixing, which is the mixing of multiple languages and scripts in a single piece of text. Code-mixing can occur at the paragraph, sentence, word, or even sub-word level. However, using non-Roman scripts like Kannada on social media can be difficult, as most keyboard layouts and keypads use the Roman alphabet. As a result, many people prefer to use the Roman script for their social media posts. This poses challenges for natural language processing tasks such as sentiment analysis and emotion detection. In this article, we propose a model for identifying the language of code-mixed text on social media. We focus on Kannada-English code-mixing, and use a combination of deep learning and traditional machine learning techniques to achieve high accuracy in our model.

To address the challenges of code-mixed text in the context of the Kannada-English language pair, we conducted a shared task for identifying the language of code-mixed text on social media. In particular, we have open-sourced a Kannada-English code-mixed dataset for word level language identification of Kannada, English, and mixed-language words written in the Roman script. The task includes classifying each word in the given text into one of six predefined categories: Kannada, English, Kannada-English, Name, Location, and Other. Among the models submitted by participants, the best performing model obtained averaged-weighted and averaged-macro F1 scores of 0.86 and 0.62, respectively.

The results of the shared task reveal the difficulty of the language identification task in code-mixed text. This difficulty is mainly due to the nature of code-mixed texts that do not follow the rules and grammar of any language. This task aims to attract the attention of researchers for word level language identification of different language pairs in code-mixed text.

Organizing Committee

Program Committee Chairs

Fazlourrahman Balouchzahi, Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico
Sabur Butt, Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico
Asha Hegde, Mangalore University, India
Noman Ashraf, Dana-Farber Cancer Institute, Harvard Medical School, United States
Shashirekha Hosahalli Lakshmaiah, Mangalore University, India
Grigori Sidorov, Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico
Alexander Gelbukh, Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico

Program Committee

Program Committee

Fazlourrahman Balouchzahi, Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico
Sabur Butt, Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico
Asha Hegde, Mangalore University, India
Noman Ashraf, Dana-Farber Cancer Institute, Harvard Medical School, United States
Shashirekha Hosahalli Lakshmaiah, Mangalore University, India
Grigori Sidorov, Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico
Alexander Gelbukh, Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico

Table of Contents

Language Identification at the Word Level in Code-Mixed Texts Using Character Sequence and Word Embedding

O. E. Ojo[1, a], **A. Gelbukh**[1, b], **H. Calvo**[1, c], **A. Feldman**[2, d],
O. O. Adebanji[1, e], **J. Armenta-Segura**[1, f]
[1]Instituto Politécnico Nacional, Centro de Investigación en Computación, CDMX, Mexico
[2]Montclair State University, USA
{[a]olumideoea, [e]olaronke.oluwayemisi}@gmail.com, [d]feldmana@montclair.edu,
{[b]gelbukh,[c]hcalvo,[f]jarmentas2022}@cic.ipn.mx

Abstract

People often switch languages in conversations or written communication in order to communicate thoughts on social media platforms. The languages in texts of this type, also known as code-mixed texts, can be mixed at the sentence, word, or even sub-word level. In this paper, we address the problem of identifying language at the word level in code-mixed texts using a sequence of characters and word embedding. We feed machine learning and deep neural networks with a range of character-based and word-based text features as input. The data for this experiment was created by combining YouTube video comments from code-mixed Kannada and English (Kn-En) texts. The texts were pre-processed, split into words, and categorized as "Kannada", "English", "Mixed-Language", "Name", "Location", and "Other". The proposed techniques were able to learn from these features and were able to effectively identify the language of the words in the dataset. The proposed CK-Keras model with pre-trained Word2Vec embedding was our best-performing system, as it outperformed other methods when evaluated by the F1 scores.

1 Introduction

Language Identification (LI), the process of automatically recognizing the language(s) present in a given document, is a key component of several text processing pipelines. The main task in LI is text classification, which is the act of assigning text to categories represented by a finite number of labels. As social media facilitate rapid information exchange and generate large amounts of text, dealing with the many different languages in documents obtained from social media users is one of the challenges in natural language processing (NLP). Code mixing is the practice of using multiple languages at once, changing the lexical and grammatical aspects of informal communication, especially on social media. Social media users around the world often combine multiple languages to express their opinions online. NLP, a branch of artificial intelligence, plays an important role in understanding the language in which humans write and speak. In some circumstances, it is very difficult to identify languages in texts collected from social media, but computational methods can be applied to automatically recognize these languages. State-of-the-art methods (Balouchzahi and Shashirekha, 2020) (Hosahalli Lakshmaiah et al., 2022) (Ojo et al., 2021) in NLP tasks apply word embedding and n-gram-based models at the character or word level for different tasks, including LI.

There are several countries where different natural languages are spoken. Language diversity has a great impact on people in marriage, sports, business, medicine, and education. In India, officially known as the Republic of India, citizens have the ability to read, write, and speak various languages. The country is one of the largest by population, and citizens can communicate in a variety of languages including Kannada and English. Code mixing, which is the practice of switching between two or more languages, is widespread in multilingual societies like India. People frequently communicate in more than one language and not always in the language in which they are addressed. The multi-language texts of users in these multilingual societies are often difficult to understand and analyze. With several understudied low-resource languages, the challenge of LI in code-mixed text is far from being solved. With this motivation, our task was to develop and evaluate models that can correctly classify labels in code-mixed Kannada and English texts.

Machine learning algorithms (Hosahalli Lakshmaiah et al., 2022) (Sidorov et al., 2014) (Ojo et al., 2020) (Kolesnikova and Gelbukh, 2010), as well as deep learning algorithms (Balouchzahi et al., 2021) (Hoang et al., 2022) (Tonja et al., 2022) (Ojo et al., 2022), have been used in many sequence

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 1 - 6
December 15-18, 2022 ©2022 Association for Computational Linguistics

labeling tasks in NLP. To identify language at the word level in Kannada-English code-mixed text, we proposed two machine learning techniques submitted to the "CoLI-Kanglish: Word-Level Language Identification in Code-mixed Kannada-English Texts" shared task (Balouchzahi et al., 2022), namely CK-Multiplex and CK-Keras. The developed models were applied to the CoLI-Kenglish dataset where the language of each word was detected and classified into a specified category.

The next section highlights the background and previous research on code mixing in social media texts. The dataset used to investigate the code mixing between English and Kannada is then introduced in Section 3. The methods applied to recognize word-level languages are discussed in Section 4. The language detection experimental results are shown in Section 5, and the conclusions and future work are presented in Section 6.

2 Background and Related Work

A number of machine learning and neural networks have been used to tackle and improve various NLP tasks, including the classification of code-mixed languages. Code mixing, according to (Muysken et al., 2000), refers to a situation in which words and grammar from two or more distinct languages are combined in a single sentence. In addition, code mixing is used while speaking two different languages at the same time. It suggests that all lexical and grammatical components indicate the act of switching languages, and that code mixing is most common in informal settings and occurs when the conversants use both languages concurrently.

(Shekhar et al., 2020) offered a method that was applied to the Facebook, Twitter, and WhatsApp dataset to identify the language of the text that has been mixed with Hindi and English. Certain sub-classes of the quantum LSTM network model have been shown to be able to accurately learn and predict language in a text on social media. The obtained results pave the way for further use of machine learning methods in quantum dynamics without relying on the precise form of the Hamiltonian.

To identify the language of Twitter data, (Ansari et al., 2021) conducted an extensive experiment using transfer learning and fine-tuning of BERT models. For language pre-training and word-level language classification, the study uses a data set consisting of code-mixed texts in Hindi, English, and Urdu. The findings demonstrate that pre-trained representations on code-mixed data perform better than their monolingual counterparts.

(Yasir et al., 2021) addresses the issue of mixed-script identification for a dataset that comprises Roman Urdu, Hindi, Saraiki, Bengali and English. RNN and word vectorization were used to train the language identification model. Furthermore, numerous model architectures were optimized, such as long short-term memory (LSTM), bidirectional LSTM, gated recurrent unit (GRU), and bidirectional gated recurrent unit (BGRU), and experimentation yielded a very good performance score. The study also looked at multilingual challenges including Roman words fused with English letters, generative spellings, and phonetic typing.

For a code-mixed text in English-Bodo-Assamese, (Kalita et al., 2021) was able to identify the language of the text at the word level. Several classification methods were applied to analyze and predict the language of text collected from Facebook. The n-gram and dictionary-based features were used to train the models on the code-mixed corpus and yielded different accuracies for the word-level language detection task.

For word-level language detection in code-mixed text, (Chittaranjan et al., 2014) developed a CRF-based system. Their method can be replicated on different languages since it takes advantage of lexical, contextual, character n-gram, and special character features. The experimental results show that the CRF-based technique performs consistently across language pairs when its performance is compared to other datasets.

To identify language boundaries at the word level, (Dutta, 2022) conducted a study using chat message datasets in mixed English-Bengali and English-Hindi languages. The author introduced a code-mixing index to evaluate the level of mixing in the corpora and evaluated the performance of the system to multiple languages.

(Jhamtani et al., 2014) proposed several techniques to learn the sequence of characters that are frequently swapped for others in standard transliterations. The authors demonstrated how these algorithms can do better than others in identifying Hindi words that correlate with the transliterated words supplied. Their distinctive experimental model for word-level language identification considers the language and part of speech of nearby words. The experimental findings indicate that the proposed model performs better in terms of accuracy than the previous methods.

To help machine learning (ML) classifiers tackle the issue of offensive language identification (OLI) in code-mixed and multi-script texts, (Balouchzahi et al.) proposed the use of relevant features of syllables and character n-grams. Three pairs of Dravidian languages, Malayalam-English, Tamil-English, and Kannada-English, were used to evaluate the performance of the proposed models. Syllable and character n-gram features performed well for code-mixed and multi-script text analysis, as shown by the results of ML classifiers.

(Mandal and Singh, 2018) developed a unique architecture for code-mixed data language tagging that uses multichannel neural networks that mix CNN and LSTM for code-mixed data word level language identification. This architecture incorporates context information. The multichannel neural network performed well in the language identification task when used with a Bi-LSTM-CRF context capture module.

3 Dataset

The words in the CoLI-Kenglish dataset, provided by the shared task organizers, were written in Kannada, English, or a combination of the two languages and are classified into six main groups: "Kannada", "English", "Mixed-language", "Name", "Location", and "Other". The data was scraped from Kannada YouTube video comments and preprocessed according to (Hosahalli Lakshmaiah et al., 2022). The unstructured texts with incomplete sentences and shortened words were code-mixed between the two languages. Two native Kannada speakers carefully tagged 19,432 unique words extracted from more than 7,000 sentences to create the CoLI-Kenglish dataset. Table 1 contains a description of the data. The test dataset includes words of unknown language. This single-label classification only allows one language to be assigned to each word, and the languages can be either "Kannada" or "English" or "Mixed-language" or "Name" or "Location" or "Other". Table 2 shows the percentage of words in each category.

4 Methodology

In different text classification tasks, numerous algorithms have been proposed and yielded promising results. The models predicted the categories of the words in the vocabulary based on the feature responses received from the vector representation of the text. Data cleansing, word segmentation, and tokenization are typically the pre-processing steps applied to the raw input text and used to train the models. The text representation transmits the pre-processed text in the form of N-gram (Cavnar et al., 1994), Bag-Of-Words (BOW) (Zhang et al., 2010), Term Frequency-Inverse Document Frequency (TF-IDF) (Peng et al., 2014), and Word2Vec representations (Mikolov et al., 2013) that the models can understand while minimizing information loss.

Word2Vec model (Mikolov et al., 2013) generates word vectors for semantic meanings using local contexts. A word vector is a fixed-length real-value vector that is used to represent any word in the corpus. Word2Vec employs two critical models: CBOW and Skip-gram. The first way entails guessing the term that is being used on the assumption that its context is understood. When the word in use is known, the latter predicts the context. The Word2Vec training approach helps the system learn vector representations of words using the structure of the neural network. The proposed techniques implement systems based on well-researched methodologies such as character ngram and Word2Vec embedding.

4.1 CK-Multiplex

The initial model used for the text classification task by the CK-Multiplex is the Random Forest Classifier (RFC). Subsequently, the multilayer perceptron (MLP) classification model was used, which has shown positive results in terms of performance.

- **Random Forest Classifier (RFC)**

  The random forest classifier is one of the supervised learning algorithms that blends ensemble learning techniques with the decision tree architecture. It can manage large data sets and can automatically balance data sets when one class is more frequent than others. RFC does not require feature scaling since it employs a rule-based approach rather than distance calculation, and non-linear factors have no impact on its performance. It is extremely stable, robust to outliers, and has a lower noise impact.

- **Multi-Layer Perceptron (MLP)**

  The multi-layer perceptron (MLP) is a feed-forward neural network that learns the associ-

Category	Tag	Description
Kannada	kn	Kannada words written in Roman script
English	en	Pure English words
Mixed-language	kn-en	Combination of Kannada and English words in Roman script
Name	name	Words that indicate name of person (including Indian names)
Location	location	Words that indicate locations
Other	other	Words not belonging to any of the above categories and words of other languages

Table 1: Description of the CoLI-Kenglish dataset

Category	Tag	% of words
Kannada	kn	43.9%
English	en	30.1%
Mixed-language	kn-en	9.3%
Name	name	4.8%
Location	location	0.7%
Other	other	11.2%

Table 2: Percentage of words per category

ations between linear and non-linear data. It has one input layer with one node (or neuron) for each input, one output layer with one node for each output, and any number of hidden layers, each with any number of nodes. The multi-layer perception uses sigmoid activation functions at each node. What makes the MLP model so potent is its ability to learn the representation in the training data, as well as its capacity to learn any mapping function and being shown to be a universal approximation method.

Character n-grams were used as a very effective feature set in both the RFC and MLP models. Character n-grams can identify a word's morphological structure, in contrast to word n-grams, which can only recognize a word and its potential neighbors. Characters n-grams are much more effective in spotting patterns than word n-grams when identifying language in text. The results of the ngram model for each language category are obtained and recorded in Table 3.

4.2 CK-Keras

The system architecture to distinguish Kannada from English at word level is built on Long Short-Term Memory (LSTM) neural network and Word2Vec embedding. LSTM networks have been at the cutting edge of sequence-to-sequence learning (Chang and Lin, 2014) (Adebanji et al., 2022). Order dependency in sequence prediction tasks can be learned with LSTM neural networks that also contain internal states that can encode context input. The LSTM network architecture can handle text as a long word or character string and incorporates feedback loops to help keep information over time. LSTM can encode internal text structures such as word dependencies and is perfectly suited for language identification. It is used for various NLP tasks such as time series, machine translation, and many others. Words were trained in the embedding layer of the LSTM model with a sequence length of 30 and a batch size of 64 in CK-Keras and then transferred to the next level with the embedding layer. The length of the sequence defines the features of the dataset.

Model	Language	Prec.	Recall	F1
RFC	en	0.80	0.84	0.82
	en-kn	0.85	0.56	0.68
	kn	0.71	0.93	0.81
	location	1.00	0.07	0.12
	name	0.73	0.23	0.35
	other	0.65	0.20	0.31
MLP	en	0.76	0.78	0.77
	en-kn	0.72	0.68	0.70
	kn	0.78	0.79	0.79
	location	0.44	0.13	0.21
	name	0.44	0.36	0.39
	other	0.47	0.48	0.48

Table 3: Performance scores for the CK-Multiplex Model on the language categories

5 Results

After applying our language identification models to the dataset, we were able to classify the words in the test set according to the categories that the

Model	Features	W.A. F1-score	M.A. F1-score	Accuracy
RFC	Character n-grams	0.71	0.51	0.74
MLP	Character n-grams	0.71	0.54	0.72
LSTM	Word2Vec embedding	0.72	0.56	0.77

Table 4: Comparison of the F1 and accuracy scores of the CK-Multiplex and CK-Keras Models (W.A. - Weighted Average, M. A. - Macro Average)

models had learned from. The languages were identified using Random Forest Classifiers and Multi-Layer Perceptron baseline models and the results are encouraging. We also used LSTM's deep learning model to learn a better feature from the text. LSTMs were further trained using random initialized word embeddings. The systems were evaluated using accuracy, recall, and F1 scores, and the results obtained are shown in Tables 3 and 4.

6 Conclusion and Future Works

In this study, a preliminary investigation was conducted to determine the language used in code mixing during interaction on social media. The vocabulary and grammar of code-mixed texts are often adapted from multiple languages, and new structures are frequently developed based on the language and usage habits of its users. We tackle the problem of language identification at the word level in code-mixed social media text containing English and Kannada languages. We use a two-step classification approach for the word-level language identification task. The embedding of character, sub-word, and word-level information can assist in the learning of meaningful correlations in words from many different languages. The LSTM recurrent neural network and Word2Vec embedding approach achieved the highest F1 score among the proposed models. In the future, we plan to use more deep learning models and text from other languages. In order to extract useful information from code-mixed texts and make code-switching systems better understand reviews, comments, inquiries, sentiments, etc., it is necessary to adequately detect and process the language of these texts.

Acknowledgements

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20211884, 20220859, and 20220553, of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

Olaronke Oluwayemisi Adebanji, Irina Gelbukh, Hiram Calvo, and Olumide Ebenezer Ojo. 2022. Sequential models for sentiment analysis: A comparative study. In *Mexican International Conference on Artificial Intelligence*, pages 227–235. Springer.

Mohd Zeeshan Ansari, MM Sufyan Beg, Tanvir Ahmad, Mohd Jazib Khan, and Ghazali Wasim. 2021. Language identification of hindi-english tweets using code-mixed bert. In *2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 248–252. IEEE.

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*.

Fazlourrahman Balouchzahi and HL Shashirekha. 2020. Mucs@ dravidian-codemix-fire2020: Saco-sentimentsanalysis for codemix text. In *FIRE (Working Notes)*, pages 495–502.

Fazlourrahman Balouchzahi, Hosahalli Lakshmaiah Shashirekha, Grigori Sidorov, and Alexander Gelbukh. A comparative study of syllables and character level n-grams for dravidian multi-script and code-mixed offensive language identification. *Journal of Intelligent & Fuzzy Systems*, (Preprint):1–11.

William B Cavnar, John M Trenkle, et al. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and* *information retrieval*, volume 161175. Las Vegas, NV.

Joseph Chee Chang and Chu-Cheng Lin. 2014. Recurrent-neural-network for language detection on twitter code-switching corpus. *arXiv preprint arXiv:1412.4314*.

Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury. 2014. Word-level language identification using crf: Code-switching shared task report of msr india system. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 73–79.

Aparna Dutta. 2022. Word-level language identification using subword embeddings for code-mixed bangla-english social media data. In *Proceedings of the Workshop on Dataset Creation for Lower-Resourced Languages within the 13th Language Resources and Evaluation Conference*, pages 76–82.

Thang Ta Hoang, Olumide Ebenezer Ojo, Olaronke Oluwayemisi Adebanji, Hiram Calvo, and Alexander Gelbukh. 2022. The combination of bert and data oversampling for answer type prediction. In *CEUR Workshop Proceedings*, volume 3119. CEUR-WS.

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*.

Harsh Jhamtani, Suleep Kumar Bhogi, and Vaskar Raychoudhury. 2014. Word-level language identification in bi-lingual code-switched texts. In *Proceedings of the 28th Pacific Asia Conference on language, information and computing*, pages 348–357.

Nayan Jyoti Kalita, Ankita Goyal Agarwala, and Jayprakash Das. 2021. Word level language identification on code-mixed english-bodo text. In *IOP Conference Series: Materials Science and Engineering*, volume 1020, page 012027. IOP Publishing.

Olga Kolesnikova and Alexander Gelbukh. 2010. Supervised machine learning for predicting the meaning of verb-noun combinations in spanish. In *Mexican International Conference on Artificial Intelligence*, pages 196–207. Springer.

Soumil Mandal and Anil Kumar Singh. 2018. Language identification in code-mixed data using multichannel neural networks and context capture. *arXiv preprint arXiv:1808.07118*.

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

Pieter Muysken et al. 2000. *Bilingual speech: A typology of code-mixing*. Cambridge University Press.

OE Ojo, A Gelbukh, H Calvo, and OO Adebanji. 2021. Performance study of n-grams in the analysis of sentiments. *Journal of the Nigerian Society of Physical Sciences*, pages 477–483.

Olumide E Ojo, Alexander Gelbukh, Hiram Calvo, Olaronke O Adebanji, and Grigori Sidorov. 2020. Sentiment detection in economics texts. In *Mexican International Conference on Artificial Intelligence*, pages 271–281. Springer.

Olumide Ebenezer Ojo, Thang Ta Hoang, Alexander Gelbukh, Hiram Calvo, Grigori Sidorov, and Olaronke Oluwayemisi Adebanji. 2022. Automatic hate speech detection using cnn model and word embedding. *Computación y Sistemas*, 26(2).

Tao Peng, Lu Liu, and Wanli Zuo. 2014. Pu text classification enhanced by term frequency–inverse document frequency-improved weighting. *Concurrency and computation: practice and experience*, 26(3):728–741.

Shashi Shekhar, Dilip Kumar Sharma, and MM Sufyan Beg. 2020. Language identification framework in code-mixed social media text based on quantum lstm—the word belongs to which language? *Modern Physics Letters B*, 34(06):2050086.

Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernández. 2014. Syntactic n-grams as machine learning features for natural language processing. *Expert Systems with Applications*, 41(3):853–860.

Atnafu Lambebo Tonja, Olumide Ebenezer Ojo, Muhammad Arif, Abdul Gafar Manuel Meque, Olga Kolesnikova, Grigori Sidorov, and Alexander Gelbukh. 2022. Cic nlp at smm4h 2022: a bert-based approach for classification of social media forum posts. In *Mining for Health Applications, Workshop & Shared Task (# SMM4H 2022)*, page 58.

Muhammad Yasir, Li Chen, Amna Khatoon, Muhammad Amir Malik, and Fazeel Abid. 2021. Mixed script identification using automated dnn hyperparameter optimization. *Computational intelligence and neuroscience*, 2021.

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1(1):43–52.

CoLI-Kanglish: Word-Level Language Identification in Code-Mixed Kannada-English Texts Shared Task using the Distilka model

Vajratiya Vajrobol
tiya101@south.du.ac.in
Institute of Informatics and Communication
University of Delhi

Abstract

Due to the intercultural demographic of online users, code-mixed language is often used by them to express themselves on social media. Language support to such users is based on the ability of a system to identify the constituent languages of the code-mixed language. Therefore, the process of language identification that helps in determining the language of individual textual entities from a code-mixed corpus is a current and relevant classification problem. Code-mixed texts are difficult to interpret and analyze from an algorithmic perspective. However, highly complex transformer-based techniques can be used to analyze and identify distinct languages of words in code-mixed texts. Kannada is one of the Dravidian languages which is spoken and written in Karnataka, India. This study aims to identify the language of individual words of texts from a corpus of code-mixed Kannada-English texts using transformer-based techniques. The proposed *Distilka* model was developed by fine-tuning the DistilBERT model using the code-mixed corpus. This model performed best on the official test dataset with a macro-averaged F1-score of 0.62 and weighted precision score of 0.86. The proposed solution ranked first in the shared task.

1 Introduction

Language identification is the process of determining the natural language that a document is written in. Automatic language identification has been widely researched for 50 years (Jauhiainen et al., 2019). However, while recognizing text in different languages might come naturally to a human reader, it is still challenging for the computer. Natural Language Processing (NLP) focuses on teaching computers to comprehend spoken and written language in a manner similar to humans. NLP research is evolving and rapidly expanding. Classification tasks such as text classification, sentiment analysis, name entity recognition, and speech recognition have been solved in the past using algorithms of NLP. Similarly, language identification, is a recent research problem that can be dealt with using NLP techniques (Shanmugalingam et al., 2018)

Focusing on language identification when it comes to a low-resource language or a mixed language, in order to identify the language could be a huge challenge. As we know, India has a rich language culture covering different geographical areas such as Hindi, Bengali, and Kannada. Kannada is spoken mainly in Karnataka, which is a southern state of India (Kumar et al., 2015). People of Karnataka read, write, and speak Kannada, but many find it difficult to use Kannada script to post comments on social media. As a result, Kannada is a low-resource language since social media users typically use Roman script or a combination of Kannada and Roman script. In this shared task, a dataset was created using Kannada YouTube comments named “CoLI-Kanglish”. (Balouchzahi et al., 2022; Shashirekha et al., 2022).

The main objective of this shared task is to create a novel method for language identification in mixed languages that consists of tokens from English, Kannada, mixed languages of Kannada and English, name, location, and other categories. In this study, the dataset is initially prepared by

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 7 - 11
December 15-18, 2022 ©2022 Association for Computational Linguistics

task organizers. Then, the data processing technique is utilized. Finally, we evaluate the model using macro-averages and weighted average scores. The following points are the contribution of our study:

- Exploratory data analysis of the dataset

- Create the Distilka (**Distil**BERT + **Ka**nnada) model based on the Transformer-based DistilBERT.

To the best of our knowledge, this is the first study that applies a DistilBERT-based model to identify mixed language in Kannada and English (CoLI-Kanglish) datasets, and the fact that it presents the best performance is highlighted.

2 Related works

Language identification has been studied for half a decade, and automatic language identification has been proposed rigorously in social media data. In 2014, Barman et al, presented an initial study on automatic language identification using Indian language code mixed in social media communication. The dataset of Bengali, Hindi, and English in Facebook comments. The authors conclude that character n-gram features, contextual information is also important, and information from dictionaries can be useful for Language Identification tasks. Apart from Facebook data, the studies also investigate Twitter data with Support Vector Machine (linear kernel), which contains bilingual tweets written in the most commonly used Iberian languages (i.e., Spanish, Portuguese, Catalan, Basque, and Galician) as well as English language. The study achieved 0.792 for macro-F1 (Pla & Hurtado, 2017).

In the same year, 2017, Transformers were introduced. The paper "Attention is all you need," describing attention mechanisms, provides context for any position in the input sequence. (Vaswani et al., 2017). Furthermore, if the input data is a natural language sentence, the transformer does not have to process one word at a time. This allows for more parallelization than a recurrent neural network and therefore reduces training times. In 2018, BERT (Bidirectional Encoder Representations from Transformers) from transformer-based technique, was developed with large amounts of pre-trained data and the ability to capture context, so BERT has become a well-known architecture since then (Devlin et al., 2018). Due to some constraints of BERT, DistilBERT has emerged to optimize the training by reducing the size of BERT and increasing the speed of BERT,while trying to retain as much performance as possible. Moreover, DistilBERT is 40% smaller than the original BERT base model, is 60% faster than it, and retains 97% of its functionality.

DistilBERT has been used in a variety of text classification tasks, including language identification. There are several papers using DistilBERT for text classification tasks, for example. Bambroo & Awasthi, 2021 proposed a fine-tuned DistilBERT on legal-domain specific corpora and discovered that this model outperformed other algorithms while also being faster at the task of legal document classification. Another study is to carry out a word-level language identification (WLLI) of Malayalam-English code-mixed data from YouTube. According to the study, DistilBERT produced the highest precision score with 91.74% in Hindi and English pairs (Thara & Poornachandran, 2021).

3 Experiments

3.1. Datasets

The CoLI-Kanglish dataset includes English and Kannada words written in Roman script and is divided into six labels: "Kannada," "English," "Mixed-language," "Name," "Location," and "Other. The CoLI-Kanglish (train dataset) contains 14,847 tokens, and there are 6 tags. Table 1 shows the number of entries in each category. In addition, the test dataset consists of 4,585 tokens without labels. The example of the dataset can be found in Table 2.

Category	Tag	Count
Kannada	kn	6,526
English	en	4,469
Mixed-Language	kn-en	1,379
Name	name	708
Location	location	102
Other	other	1,663

Table 1. The description and samples of tokens in CoLI-Kanglish Dataset

Word	Tag
hegilla	kn
staying	en
aparictarannu	en-kn
kamal	name
bangalore	location
mamao	other

Table 2. The example of CoLI-Kanglish Dataset in each tag

3.2 DistilBERT

A distilled version of BERT that is smaller, quicker, less expensive, and lighter was proposed by Sanh et al., 2019. DistilBERT is a BERT base-trained transformer model that is compact, quick, affordable, and light. It runs 60% faster with 40% fewer parameters than BERT-base-uncased while maintaining over 95% of BERT's performance as measured by the GLUE language understanding benchmark. When compared to other models, DistilBERT produces the quickest results with 106 seconds (Bambroo & Awasthi, 2021).

3.3 Distilka model

Distilka is the fine-tuned model in the mix-language Kannada and English identification tasks by using DistilBERT-based categorization with 6 labels such as "Kannada," "English", "Mixed-language", "Name", "Location", and "Other". This model can be downloaded from the Hugging Face Hub (https://huggingface.co/tiya1012/distilka_applied), and the framework of this study is illustrated in Fig 1.

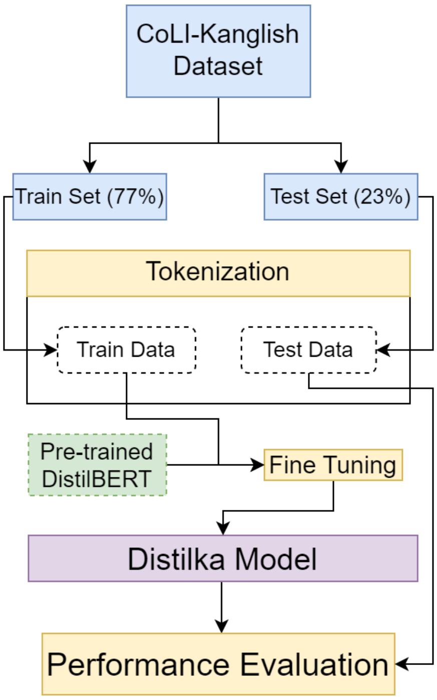


Fig 1. Framework of Model Development for Language Identification using CoLI-Kanglish Dataset

4 Experiments and Results

4.1 Experimental setting

The notebook ran on Google Colab Pro with Python 3 installed, and the model was fine-tuned. We set the learning rate at 2e-5, the maximum sequence length at 512, and the gradient accumulation steps at 1 and batch size was set at 6 as shown in Table 3. The optimal results were obtained through a comparative study which is shown in Table 4.

Parameters	Values
Maximum sequence length	512
Learning rate	2e-5
Accumulated gradient steps	1
Batch Size	6

Table 3. Model Hyperparameters

4.2 Results and Discussion

As we can see from Table 4, it represents the weighted and macro precision and F1-score. In this shared task, ranking will be finalized using macro F1 score. Since the Distilka model was trained and learned from the DistilBERT-based-cased, its macro F1-score is 0.62. There are several factors that contribute to the best performance of DistilBERT; for instance, DistilBERT is pretrained on the same data as BERT, which is BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia. Although this dataset contains Kannada language, it has been written in English. Furthermore, the model can learn better if more datasets have been trained. In the fine-tuning period, language tag data is used based on the ability of the language model to improve the performance of the downstream tasks. This enables DistilBERT to achieve cutting-edge results in written Kannada-English language benchmarks.

Model	Precision (weighted)	F1-score (weighted)	F1-score (Macro)
Distilka	0.87	0.86	0.62

Table 4. Results of Distilka model

5 Conclusion

In this paper, we describe our proposed method for the shared task of word-level language identification in code-mixed Kannada-English dataset. On comparing with the performance metrics of other solutions based on DistilBERT, that were developed for this shared task, it was found out that the Distilka model performed significantly better with a macro-averaged F1-score of 0.62. The proposed model secured the first rank in the shared task. In future work, we will try to adjust the parameters of the new model in order to improve its performance significantly. Future work will include further application of language identification tasks to several low-resource languages.

References

Purbid Bambroo and Aditi Awasthi. 2021. LegalDB: Long DistilBERT for Legal Document Classification. In *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–4. IEEE.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code Mixing: A Challenge for Language Identification in the Language of Social Media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 13–23, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic Language Identification in Texts: A Survey. *Journal of Artificial Intelligence Research*, 65.

K. M. Anil Kumar, N. Rajasimha, Manovikas Reddy, A. Rajanarayana, and Kewal Nadgir. 2015. Analysis of users' Sentiments from Kannada Web Documents. *Procedia Computer Science*, 54:247–256.

Ferran Pla and Lluís-F. Hurtado. 2017. Language identification of multilingual posts from Twitter: a case study. *Knowledge and Information Systems*, 51(3):965–989.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Kasthuri Shanmugalingam, Sagara Sumathipala, and Chinthaka Premachandra. 2018. Word Level Language Identification of Code Mixing Text in Social Media using NLP. In *2018 3rd International Conference on Information Technology Research (ICITR)*, pages 1–5. IEEE.

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. *19th International Conference on Natural Language Processing Proceedings*.

H L Shashirekha, F Balouchzahi, M D Anusha, and G Sidorov. 2022. CoLI-Machine Learning Approaches for Code-mixed Language Identification at the Word Level in Kannada-English Texts.

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):94.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need.

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 ©2022 Association for Computational Linguistics

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 ULMFiT (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 ULMFiT 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.

Transformer-based Model for Word Level Language Identification in Code-mixed Kannada-English Texts

Atnafu Lambebo Tonja[1], Mesay Gemeda Yigezu[2], Olga Kolesnikova[3], Moein Shahiki Tash[4], Grigori Sidorov[5], Alexander Gelbukh[6]

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC),
Mexico City, Mexico
{ [1]alambedot2022, [2]mgemedak2022, [5]sidorov, [6]gelbukh}@cic.ipn.mx
{ [3]kolesolga, [4] moein.tash}@gmail.com

Abstract

Using code-mixed data in natural language processing (NLP) research currently gets a lot of attention. Language identification of social media code-mixed text has been an interesting problem of study in recent years due to the advancement and influences of social media in communication. This paper presents the Instituto Politécnico Nacional, Centro de Investigación en Computación (CIC) team's system description paper for the CoLI-Kanglish shared task at ICON2022. In this paper, we propose the use of a Transformer based model for word-level language identification in code-mixed Kannada English texts. The proposed model on the CoLI-Kenglish dataset achieves a weighted F1-score of 0.84 and a macro F1-score of 0.61.

1 Introduction

In recent years, language identification of social media text has been a fascinating research topic (Ansari et al., 2021). Social media platforms have become more integrated in this digital era and have impacted various people's perceptions of networking and socializing (Tonja et al., 2022c). This influence allowed different users to communicate via various social media platforms using a mix of texts. NLP technology has advanced rapidly in many applications, including machine translation (Tonja et al., 2022b; Yigezu et al., 2021; Tonja et al., 2021), abusive comment detection (Balouchzahi et al., 2022b), fake news detection(Arif et al., 2022; Truică et al., 2022), aggressive incident detection (Tonja et al., 2022a), hope speech detection (Gowda et al., 2022), and others. However, numerous tools have not yet been created for languages with limited resources or languages with code-mixed data.

Code-mixing is the use of linguistic units—words, phrases, and clauses—at the sentence or word level from various languages. In casual communication, such as social media, it is typically seen. We have access to an enormous amount of code-mixed data because of the various social media platforms that allow individuals to communicate (Sutrisno and Ariesta, 2019). As a result, automatic language recognition at the word level has become an essential part of analyzing noisy content in social media. It would help with the automated analysis of content generated on social media. Currently, in the area of NLP, different researchers are developing different NLP applications in code-mixed datasets. Some of the applications are code-mixed sentiments analysis (Balouchzahi et al., 2021b), code-mixed offensive language identification (Balouchzahi et al., 2021a), etc. We took part in the **Kanglish shared task** (Balouchzahi et al., 2022a), which aims to identify language at the word level from code-mixed data for Kannada-English texts. For word-level code-mixed language identification tasks, we used Transformer -based (Vaswani et al., 2017) pre-trained language models (PLMs). Our transformer-based model consists of BERT (Devlin et al., 2018) and its three variants. We used PLMs and LSTM models for this word-level language identification task.

This paper discusses a Transformer-based model for word-level language identification in code-mixed Kannada-English texts for the Kanglish shared task. The paper is organized as follows: Section 2 describes past work related to this study, section 3 gives an overview of the dataset and its statistics, section 4 explains the methodology adopted in this study including the algorithms, section 5 emphasizes on the experimental results and descriptions. Finally, Section 6 concludes the paper.

2 Related Work

Currently, solving NLP problems in code-mixed data is getting attention from many researchers. For word-level language identification in code-

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 18 - 24
December 15-18, 2022 ©2022 Association for Computational Linguistics

mixed text, different researchers have suggested various models. Chittaranjan et al. (2014) proposed a Conditional Random Fields (CRF)- based system for word level language identification of code-mixed text for four language pairs, namely, English-Spanish (En-Es), English-Nepali (En-Ne), English-Mandarin (En-Cn), and Standard Arabic-Arabic (Ar-Ar) dialects. The authors explored various token levels and contextual features to build an optimal CRF using the provided training data. The proposed system performed more or less consistently, with accuracy ranging from 80% to 95% across four language pairs.

Gundapu and Mamidi (2020) also proposed a CRF based model for word-level language identification in English-Telugu code-mixed data. The authors used feature extraction as the main task for the proposed model. They used POS-tags, length of the word, prefix and suffix of focus word, numeric digit, special symbol, capital letter, and character N-grams (Uni-, Bi-, Trigrams of words) as features. The proposed CRF-based model had an F1-score of 0.91.

A Support Vector Machines (SVM)-based model for word level language identification of Tamil-English code-mixed text in social media is proposed by Shanmugalingam et al. (2018). The authors used dictionaries, double consonants, and term frequency to identify features. The proposed SVM model with a linear kernel gave 89.46% accuracy for the language identification system for Tamil-English code-mixed text at the word level.

Ansari et al. (2021) proposes transfer learning and fine-tuning BERT models for language identification of Hindi-English code-mixed tweets. The authors used data from Hindi-English-Urdu code-mixed text for language pre-training and Hindi-English code-mixed for subsequent word-level language classification. The authors first pre-trained Hindi-English-Urdu code-mixed text using BERT and fine-tuned the trained model in downstream Hindi-English code-mixed word-level language classification. Their proposed model for Hindi-English code-mixed language identification, both pre-training and fine-tuning with code-mixed text, gives the best F1-score of 0.84 as compared to their monolingual counterparts.

3 Data

During the experimental phase, we used the CoLI-Kenglish dataset (Hosahalli Lakshmaiah et al., 2022) which consists of English and Kannada words in Roman script and are grouped into six major categories, namely, Kannada (kn), English (en), Mixed-language (en-kn), Name, Location and Other. Table ?? shows some samples from the dataset used for training.

	word	**tag**
0	anusthu	kn
1	woww	en
2	staying	en
3	near	en
4	hostel	en
5	confirmed	en
6	faith	en
7	linked	en
8	gtila	kn
9	germany	en

Table 1: Training samples

3.1 Dataset Statistics

Figures 1 and 2 depict the training and test data distribution statistics with their assigned tags. The training dataset is slightly imbalanced: 43.9% of the words were labeled as *kn*, 30% were labeled as *en*, 9.28% were labeled as *en-kn*, 4.76% were labeled as *name*, 0.68% were labeled as *location* and 11.2% were labeled as other. This shows that approximately 73% of the training dataset was labeled as *kn* and *en*. Similarly, in the test dataset, words tagged as *en* and *kn* take a higher number than the rest of the dataset.

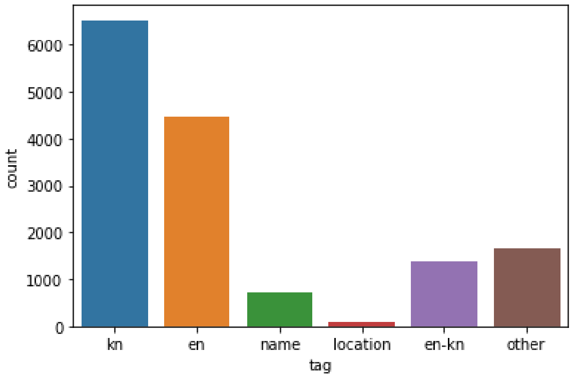


Figure 1: Training data distribution with tags

4 Methodology

This section presents a description of the data preprocessing, methodology, and models used in this

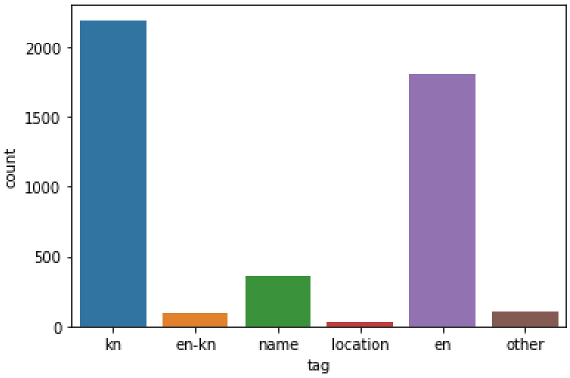


Figure 2: Test data distribution with tags

work. We used Transformer based pre-trained language models (PLMs) with the combination of the LSTM model for word level language identification in Kannada-English code-mixed text. We used PLMs in the embedding layer of the LSTM model layer.

4.1 Pre-processing

Pre-processing is one of the preliminary steps in training NLP tasks, with the aim of making the training data suitable during the training phase. The dataset provided by the organizers for this task has passed the basic pre-processing steps, and we carried out one pre-processing step to prepare the training data during the experimental phase. We applied label encoding to tags, to convert the tags into a numeric form. As discussed in section 3, the dataset contains six tags (*kn, en, en-kn, name, location* and *other*). We converted these tags into numeric values using one-hot encoding.

4.2 Proposed Experimental Architecture

Figure 3 shows the experimental architecture of our Transformer-based model for word level language identification in code-mixed Kannada-English texts. As shown in Figure 3, our experimental architecture consists of five steps:

- **Step 1** - preparing labelled data for training, the data set contains **words** and their **tags** as discussed in section 3.

- **Step 2** - we converted the tags into a numeric machine-readable form.

- **Step 3** - after label encoding the representation for each token is fed to transformer layers to obtain contextualized tokens using PLMs.

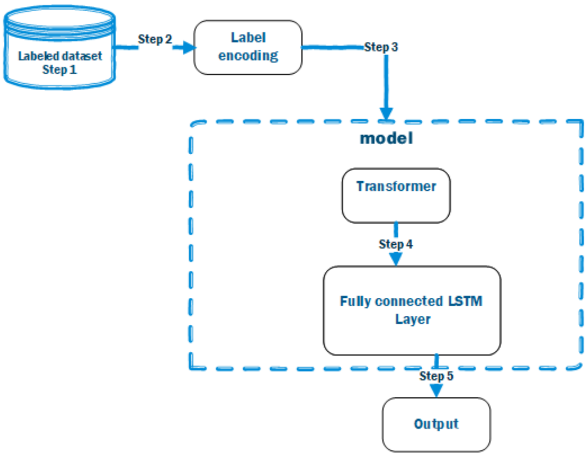


Figure 3: Experimental architecture for Transformer-based model for word level language identification in code-mixed Kannada-English texts

- **Step 4** - the embeddings obtained in step-3 are fed into LSTM model to obtain their corresponding language tag.

We used the following pre-trained language models (PLMs) in the embedding layer of the LSTM model for our experiment.

- BERT (Devlin et al., 2018) - stands for Bidirectional Encoder Representations from Transformers. As the name suggests, it is a way of learning representations of a language that uses a transformer, specifically, the encoder part of the transformer.

- mBERT (Devlin et al., 2018) - is a Multilingual BERT, it provides sentence representations for 104 languages, which are useful for many multi-lingual tasks. Previous work probed the cross-linguality of mBERT using zero-shot transfer learning on morphological and syntactic tasks.

- XLM-R (Conneau et al., 2019) - uses self-supervised training techniques to achieve state-of-the-art performance in cross-lingual understanding, a task in which a model is trained in one language and then used with other languages without additional training data.

- RoBERTa (Liu et al., 2019) is a self-supervised transformers model that was trained on a large corpus of English data. This means it was pre-trained on raw texts only, with no human labeling in any way (which is why it can use lots of publicly available data) and an automatic process to generate inputs and labels from those texts.

Table 2 shows models used in our experiments and their parameters.

5 Experiments and Results

This section presents the description of the experimental setups, training parameters, results, and analysis. We conducted four experiments by replacing embedding layers with different pre-trained language models, the results are presented in section 5.2.

5.1 Experiments

We used Google colab [1] for GPU support with the Python programming language. Sci-kit-learn [2] and Keras [3] (with TensorFlow backend) were utilized for the LSTM model, for PLMs we used Hugging Face [4] transformer libraries. We used PLMs for embedding and the LSTM model as the classifier, To optimize the model, we used an Adam optimizer with a batch size of 64 and a learning rate of 0.0001. We used the maximum number of epochs of 30, with early stopping based on the performance of the validation set. We also used a dropout of 0.2 to regularize the model.

We added a batch normalization layer to speed up training, and make learning easier, and a fully-connected output layer with a SoftMax function so that a probabilistic output of all tags for language identification would be produced. For further information, all the parameters and their summaries are depicted in Figure 4. Figure 4 shows our proposed model summary for word-level language identification in code-mixed Kannada-English texts.

5.2 Results

Table 3 depicts the overall results (official) of four experiments conducted in this work. From four experiments, using *bert-base-uncased* in the embedding layer with the LSTM model out-performs other pre-trained languages models used in the embedding layer with the LSTM model with a weighted score of 0.85 precision, 0.84 recall, 0.84 F1-scores and a micro score of 0.62 precision, 0.62 recall, 0.61 F1-scores.

The official rank of the top three teams participating in the shared task of word-level language identification in code-mixed Kannada-English texts is shown in Table 4. As shown in Table 4 our model ranked second in overall results among all participant teams.

Figures 5 and 6 display the training and, validation losses, training, and validation accuracy of the BERT-based approach for code-mixed language identification tasks. It is seen that the BERT-based model's training loss decreases and stabilizes at a specific point, but the validation loss is not as stable as the training loss. This shows that the more specialized the model becomes with training data, the worse it is able to generalize to new data, resulting in an increase in generalization error.

The above result demonstrates that transformer-based models can give promising results when applied to NLP tasks like word-level language identification in code-mixed texts without considering any linguistic features.

6 Conclusion

In this paper, we explored the application of BERT-based pre-trained language models to identify languages at the word level in code-mixed data for Kannada-English texts. Pre-trained models with a combination of the LSTM model and a BERT-based model outperformed the others and have shown promising results in identifying languages in code-mixed Kannada-English texts. Our team achieved the second place in CoLI-Kanglish: word-level language identification in the code-mixed Kannada-English texts competition.

Acknowledgements

The work was done with partial support from the Mexican Government through the grant A1S-47854 of CONACYT, Mexico, grants 20220852, 20220859, and 20221627 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

[1] https://colab.research.google.com/

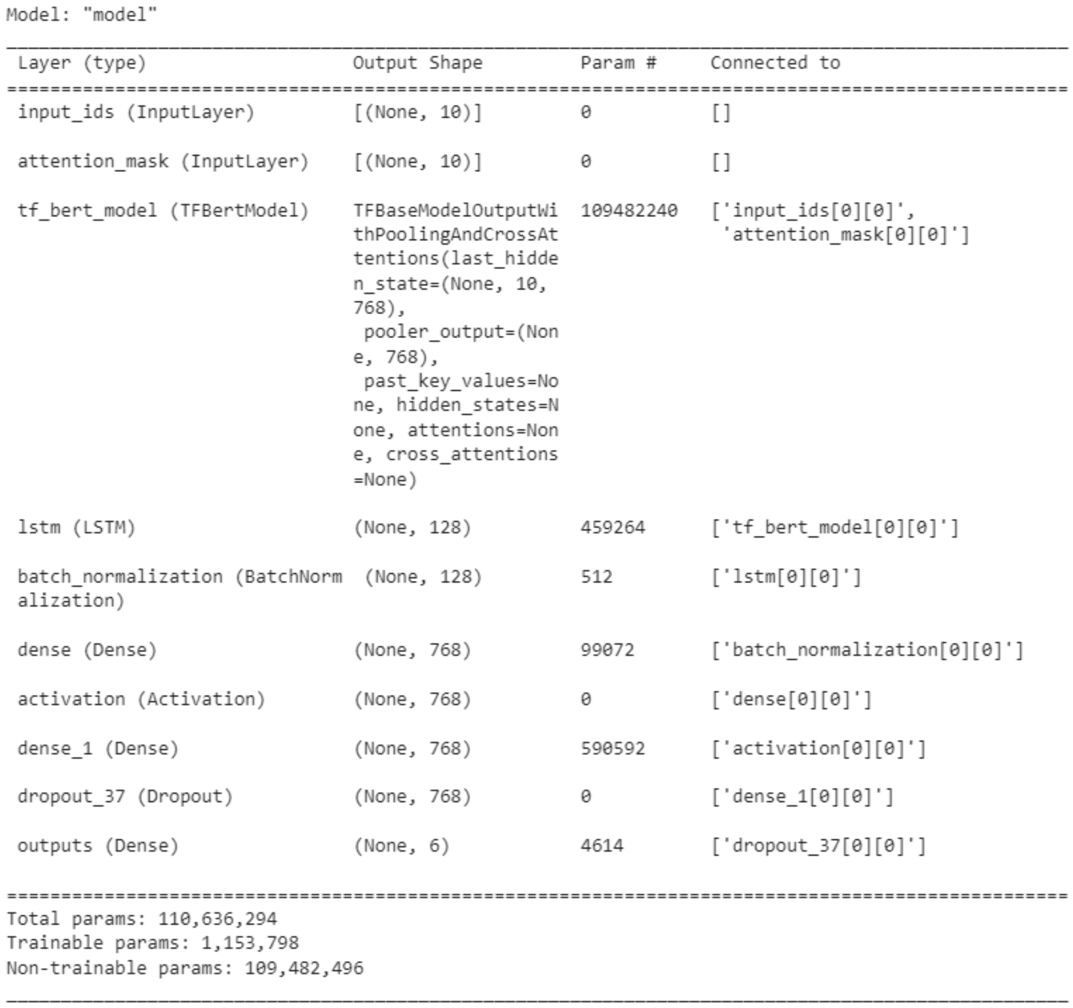
[2] https://scikit-learn.org/stable/

[3] https://keras.io/

[4] https://huggingface.co/

Model	Transformer blocks	Hidden layer size	Self-attention heads	#Parameters
bert-base-uncased	12	768	12	110M
bert-base-multilingual-uncased	12	768	12	110M
xlm-roberta-large	24	1024	16	355M
roberta-base	12	768	12	110M

Table 2: Transformers used in this paper and their parameters



```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_ids (InputLayer)         [(None, 10)]         0           []

 attention_mask (InputLayer)    [(None, 10)]         0           []

 tf_bert_model (TFBertModel)    TFBaseModelOutputWi  109482240   ['input_ids[0][0]',
                                thPoolingAndCrossAt               'attention_mask[0][0]']
                                tentions(last_hidde
                                n_state=(None, 10,
                                768),
                                 pooler_output=(Non
                                e, 768),
                                 past_key_values=No
                                ne, hidden_states=N
                                one, attentions=Non
                                e, cross_attentions
                                =None)

 lstm (LSTM)                    (None, 128)          459264      ['tf_bert_model[0][0]']

 batch_normalization (BatchNorm  (None, 128)         512         ['lstm[0][0]']
 alization)

 dense (Dense)                  (None, 768)          99072       ['batch_normalization[0][0]']

 activation (Activation)        (None, 768)          0           ['dense[0][0]']

 dense_1 (Dense)                (None, 768)          590592      ['activation[0][0]']

 dropout_37 (Dropout)           (None, 768)          0           ['dense_1[0][0]']

 outputs (Dense)                (None, 6)            4614        ['dropout_37[0][0]']

==================================================================================================
Total params: 110,636,294
Trainable params: 1,153,798
Non-trainable params: 109,482,496
__________________________________________________________________________________________________
```

Figure 4: Proposed model summary

Model	Weighted Score			Macro score		
	P	R	F1-score	P	R	F1-score
bert-base-multilingual-uncased	0.83	0.82	0.82	0.62	0.57	0.57
xlm-roberta-large	0.84	0.85	0.84	0.64	0.59	0.61
roberta-base	0.83	0.8	0.81	0.63	0.55	0.52
bert-base-uncased	**0.85**	**0.84**	**0.84**	**0.62**	**0.62**	**0.61**

Table 3: Performance of our models on the test set (official results)

Rank	Team name	Weighted Score			Macro score		
		P	R	F1-score	P	R	F1-score
1	tiya1012	0.87	0.85	0.86	0.67	0.61	0.62
2	**Our team**	**0.85**	**0.84**	**0.84**	**0.62**	**0.62**	**0.61**
2	Habesha	0.85	0.83	0.84	0.66	0.6	0.61
3	lidoma	0.83	0.83	0.83	0.64	0.56	0.58

Table 4: Official rank of top 3 teams

References

Mohd Zeeshan Ansari, MM Sufyan Beg, Tanvir Ahmad, Mohd Jazib Khan, and Ghazali Wasim. 2021. Language identification of hindi-english tweets using code-mixed bert. In *2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 248–252. IEEE.

Muhammad Arif, Atnafu Lambebo Tonja, Iqra Ameer, Olga Kolesnikova, Alexander Gelbukh, Grigori

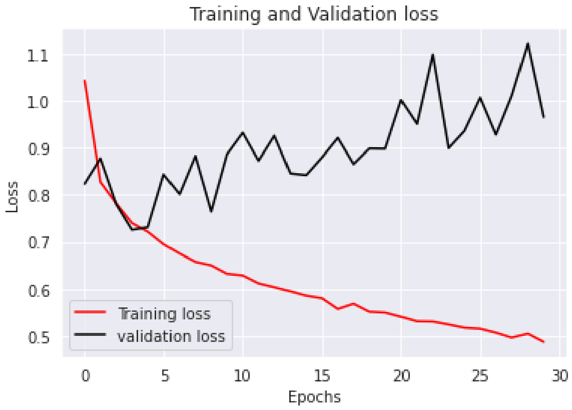


Figure 5: Training and validation loss of BERT based approach

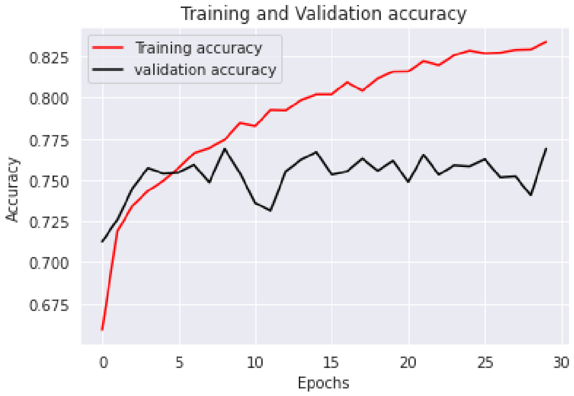


Figure 6: Training and validation accuracy of BERT based approach

Sidorov, and AG Meque. 2022. Cic at checkthat! 2022: multi-class and cross-lingual fake news detection. *Working Notes of CLEF*.

F Balouchzahi, S Bashang, G Sidorov, and HL Shashirekha. 2021a. Comata oli-code-mixed malayalam and tamil offensive language identification. In *Working Notes of FIRE 2021-Forum for Information Retrieval Evaluation (Online). CEUR*.

Fazlourrahman Balouchzahi, Sabur Butt, Asha Hagde, Noman Ashraf, Shashirekha Hosahalli Lakshmaiah, Grigori Sidorov, and Alexander Gelbukh. 2022a. 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*.

Fazlourrahman Balouchzahi, Anusha Gowda, Hosahalli Shashirekha, and Grigori Sidorov. 2022b. Mucic@ tamilnlp-acl2022: Abusive comment detection in tamil language using 1d conv-lstm. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 64–69.

Fazlourrahman Balouchzahi, Hosahalli Lakshmaiah Shashirekha, and Grigori Sidorov. 2021b. Cosad-code-mixed sentiments analysis for dravidian languages. In *CEUR Workshop Proceedings*, volume 3159, pages 887–898. CEUR-WS.

Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury. 2014. Word-level language identification using crf: Code-switching shared task report of msr india system. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 73–79.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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*.

Anusha Gowda, Fazlourrahman Balouchzahi, Hosahalli Shashirekha, and Grigori Sidorov. 2022. Mucic@ lt-edi-acl2022: Hope speech detection using data re-sampling and 1d conv-lstm. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 161–166.

Sunil Gundapu and Radhika Mamidi. 2020. Word level language identification in english telugu code mixed data. *arXiv preprint arXiv:2010.04482*.

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*.

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*.

Kasthuri Shanmugalingam, Sagara Sumathipala, and Chinthaka Premachandra. 2018. Word level language identification of code mixing text in social media using nlp. In *2018 3rd International Conference on Information Technology Research (ICITR)*, pages 1–5. IEEE.

Bejo Sutrisno and Yessika Ariesta. 2019. Beyond the use of code mixing by social media influencers in instagram. *Advances in Language and Literary Studies*, 10(6):143–151.

Atnafu Lambebo Tonja, Muhammad Arif, Olga Kolesnikova, Alexander Gelbukh, and Grigori Sidorov. 2022a. Detection of aggressive and violent incidents from social media in spanish using pre-trained language model. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022), CEUR Workshop Proceedings. CEUR-WS. org*.

Atnafu Lambebo Tonja, Olga Kolesnikova, Muhammad Arif, Alexander Gelbukh, and Grigori Sidorov. 2022b. Improving neural machine translation for low resource languages using mixed training: The case of ethiopian languages. In *Mexican International Conference on Artificial Intelligence*, pages 30–40. Springer.

Atnafu Lambebo Tonja, Olumide Ebenezer Ojo, Mohammed Arif Khan, Abdul Gafar Manuel Meque, Olga Kolesnikova, Grigori Sidorov, and Alexander Gelbukh. 2022c. Cic nlp at smm4h 2022: a bert-based approach for classification of social media forum posts. In *Proceedings of The Seventh Workshop on Social Media Mining for Health Applications, Workshop & Shared Task*, pages 58–61.

Atnafu Lambebo Tonja, Michael Melese Woldeyohannis, and Mesay Gemeda Yigezu. 2021. A parallel corpora for bi-directional neural machine translation for low resourced ethiopian languages. In *2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, pages 71–76. IEEE.

Ciprian-Octavian Truică, Elena-Simona Apostol, and Adrian Paschke. 2022. Awakened at checkthat! 2022: fake news detection using bilstm and sentence transformer. *Working Notes of CLEF*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Mesay Gemeda Yigezu, Michael Melese Woldeyohannis, and Atnafu Lambebo Tonja. 2021. Multilingual neural machine translation for low resourced languages: Ometo-english. In *2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, pages 89–94. IEEE.

Word Level Language Identification in Code-mixed Kannada-English Texts using traditional machine learning algorithms

M. Shahiki Tash, Z. Ahani, A.L. Tonja, M. Gemeda, N. Hussain and **O. Kolesnikova**
Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC)
Corresponding: `moein.tash@gmail.com`

Abstract

Language Identification at the Word Level in Kannada-English Texts. This paper describes the system paper of CoLI-Kanglish 2022 shared task. The goal of this task is to identify the different languages used in CoLI-Kanglish 2022. This dataset is distributed into different categories including Kannada, English, Mixed-Language, Location, Name, and Others. This Code-Mix was compiled by CoLI-Kanglish 2022 organizers from posts on social media. We use two classification techniques, KNN and SVM, and achieve an F1-score of 0.58 and place third out of nine competitors.

1 Introduction

Nowadays, it is impossible to find somebody who doesn't use social media or smartphones. This prompts us to identify a new difficulty for people who have used social media. The following difficulties are just one example of the many tasks are performed in Natural Language Processing (NLP) to address various issues for instance, fake news(Arif et al., 2022), machine translation detection, sentiment analysis and language identification(Yigezu et al., 2021).

Identification language for mixed languages is the major challenge. Many users want an easy way to construct sentences, or employ habitual expressions. They try to write in a combination of two or three different languages, which leads the creation of Code-Mix data(Balouchzahi et al., 2022b). User-generated content like web articles, tweets, and message boards frequently contain code-mix text, yet majority of the language ID models in use today have been ignored. As observed in these English-Hindi examples, code-mixing entails language changes inside and across constituents.

[NP aapki profile photo] [V P pyari hai]

Your profile photo is lovely

In many areas, such as those where Hindi and English speakers coexist, code-mixing is the norm. As many as 17% of Facebook posts from India are code-mixed (Bali et al., 2014) and 3.5% from tweets (Rijhwani et al., 2017).

Nearly all social media networks where people speak several languages are Code-Mix. For instance, in a nation like India, where there are more than a dozen different languages with various alphabets, you may easily locate a Mix-Code of English and Indian languages if you check the posts on Facebook or Twitter that are linked to garments or related to shopping (Balouchzahi et al., 2022a). Because of their extensive range, code mixes cannot be adequately described in a finite number of words. Code-Mixing may contain a variety of words including words that combine the alphabets of two languages that identify an area, a person or a place, and different situations.

We will now discuss the classification system we utilized in this paper and also TF-IDF vectorizer.One of the effective supervised machine learning techniques that we may utilize for both regression and data classification is called the Support Vector Machine (SVM). Finding the hyperplane in an N-dimensional space that clearly classifies the data points is the objective of an SVM. (Ekbal and Bandyopadhyay, 2008). This means that the decision boundry line between the data points that fall into a category and those that do not is drawn clearly by the algorithm. Almost all data that is encoded as a vector is suitable for this technique. If it create a good vector from our data, we can use SVM to find good results (Tonja et al., 2022). Although KNN can be used just like SVM for both classification and regression issues, it is the primary application in classification. This algorithm stores all the data and can classify a new data point based on similarities.

This method places the new instance into the column that is more comparable to the available categories and makes the assumption that the new data is linked to the available items (Nongmeika-

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 25 - 28
December 15-18, 2022 ©2022 Association for Computational Linguistics

pam et al., 2017). As it encounters new data, this algorithm simply stores the data set during training and then classifies it into a group that is roughly similar to the present data. The TF-IDF statistic gives keywords that can be used to identify or categorize particular documents by demonstrating the relevance of certain keywords to a given set of documents (Gautam and Kumar, 2013).

2 Task description and Datasets

Language Identification (LI) is the process of automatically recognizing the languages used in a text. Kannada is one of the Dravidian languages that make up India's rich linguistic legacy and is used as the official language of the state of Karnataka. Karnataka residents can read, write, and speak Kannada, yet many find it challenging to use the language while posting messages or comments on social media.

Language identification is the process of automatically recognizing the languages used in a given text because code-mixing is one of the most challenging subjects in Natural Language Processing (NLP). The goal of the current investigation is to identify the language of the words.

As part of this work, we must determine which words are of English, Kannada, and mixed languages. The CoLI-Kenglish dataset consists of Kannada and English words written in Roman script and is divided into six main categories: "Kannada," "English," "Mixed-language," "Name," "Location," and "Other." Participants are asked to submit their methods in the Kanglish shared task, which requires that each word be recognized and categorized in one of these categories (Hosahalli Lakshmaiah et al., 2022).

3 Related Work

Language identification in social media texts is difficult because of things like social media content that has been code-mixed, using one alphabet to write in two languages at this point. Chakravarthi et al. (2021) proposed a code that combines Dravidian data in Kannada, Malayalam, and Tamil. Bohra et al. (2018) extend a Twitter data collection that include Hinglish data. They provided primary experiment findings with an accuracy of 0.71 using classifiers Support Vector Machine (SVM) and Random Forest (RF) with n-grams and lexicon-based features (Chakravarthi et al., 2020b,a). Sentiment Analysis for Dravidian Languages in Code-Mixed Data was a shared task in Dravidian-Code-Mix-FIRE2020 established by Kanwar et al. (2020). Researchers had submitted a variety of models, and they used the under sampling technique from Tomek (1976) to train some machine learning classifiers with various syntax-based n-gram features. The linear regression classifier with word and char n-gram features produced positive results with average weighted F1-scores of 0.71 and 0.62.

4 Methods

In this study, we employed standard machine learning algorithms for language identification. For this task, we used two different classifiers, including (i) support vector machines and (ii) k-nearest neighbors. We also used N-gram TF-IDF word and character features for vectorization. On each of these classifiers and this vectorization, we make a comment. For this task, we submit 5 runs, and the outcome varies each time.

4.1 Feature Engineering:

For this model, we used TF-IDF Vectorizer from the Sklearn module to extract char n-grams in the range of distinct pre-processed text data that are ready as word frames (1, 2). In Table 1 we lists the quantity of tasks, test sets, data sets, and category and tag values.

Table 1: Code-mixing language categories with test- and training-set counts

Task	Category	Tag	Number of Test-set	Number of Train-set
Task1	Kannada	kn	4585	14847
	English	en		
	Mixed-language	Kn-en		
	Name	name		
	Location	location		
	Other	other		

4.2 Model Construction:

These two classifiers were employed for the training set of data. For this challenge, we had 5 Runs. K-nearest neighbors (KNN) was employed for the first four runs. In order to vectorization, we just employed TF-IDF while using alternative parameters for support vector machines. Table 2 contains a list of all the parameters we utilized for each Run.

Table 2: Parameters that used in KNN,SVM,TF-IDF

Name of classifier/ vectorizer	Parameter1	Parameter2	Parameter3	Parameter4
KNN	n_neighbors=6	metric= 'manhattan'	p=2	weights= 'distance'
SVM	C=1.0	kernel='poly'	degree=3	gamma= 'scale'
TF-IDF	analyzer='char_wb'	ngram_range=(1,2)	min_df=0	norm= 'l1'

5 Experiments and Results

We demonstrate our experiment with text data that was gathered from YouTube. Each language pair's word should be categorized into one of the six groups shown in Table 1. The suggested method w e used 14847 data for training and 4585 data for testing and we applied The purpose of the weighted average F1-score is assessment. We have displayed the number of errors made by the KNN algorithm during four runs in Table 3. Additionally, Table 4 displays the number of mistakes produced by the SVM algorithm in a single run. It is important to note that TF-IDF is used by both algorithms. As seen in Table 3, the weighted average F1-score increased and was able to rise in each run by modifying the KNN's parameters.

Table 3: Results with using KNN classifier

	Weighted			Macro		
	Precision	Recall	F1-score	Precision	Recall	F1-score
RUN1	0.78	0.79	0.77	0.63	0.43	0.47
RUN2	0.8	0.8	0.79	0.61	0.5	0.53
RUN3	0.83	0.83	0.83	0.65	0.53	0.56
RUN4	0.83	0.83	0.83	0.64	0.56	0.58

Table 4: Results with using SVM classifier

	Weighted			Macro		
	Precision	Recall	F1-score	Precision	Recall	F1-score
RUN5	0.78	0.79	0.77	0.63	0.43	0.47

6 Conclusion

This study shows how different languages may be identified in code-mix data using a classifier that uses two algorithms, KNN and SVM. The first technique produces better results, with the best weighted average F1-score 0.58.

Acknowledgement

The work was done with partial support from the Mexican Government through the grant A1S-47854 of CONACYT, Mexico, grants 20220852, 20220859, and 20221627 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

Muhammad Arif, Atnafu Lambebo Tonja, Iqra Ameer, Olga Kolesnikova, Alexander Gelbukh, Grigori Sidorov, and AG Meque. 2022. Cic at checkthat! 2022: multi-class and cross-lingual fake news detection. *Working Notes of CLEF*.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook. In *Proceedings of the first workshop on computational approaches to code switching*, pages 116–126.

Fazlourrahman Balouchzahi, Sabur Butt, Asha Hagde, Noman Ashraf, Shashirekha Hosahalli Lakshmaiah, Grigori Sidorov, and Alexander Gelbukh. 2022a. 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*.

Fazlourrahman Balouchzahi, Sabur Butt, Grigori Sidorov, and Alexander Gelbukh. 2022b. CIC@LT-EDI-ACL2022: Are transformers the only hope? hope speech detection for Spanish and English comments. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 206–211, Dublin, Ireland. Association for Computational Linguistics.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of Hindi-English code-mixed social media text for hate speech detection. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 36–41, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P McCrae. 2020a. A sentiment analysis dataset for code-mixed malayalam-english. *arXiv preprint arXiv:2006.00210*.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John P. McCrae, and Elizabeth Sherly. 2021. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on*

Speech and Language Technologies for Dravidian Languages, pages 133–145, Kyiv. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P McCrae. 2020b. Overview of the track on sentiment analysis for dravidian languages in code-mixed text. In *Forum for information retrieval evaluation*, pages 21–24.

Asif Ekbal and Sivaji Bandyopadhyay. 2008. Bengali named entity recognition using support vector machine. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Jyoti Gautam and Ela Kumar. 2013. An integrated and improved approach to terms weighting in text classification. *International Journal of Computer Science Issues (IJCSI)*, 10(1):310.

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*.

Nikita Kanwar, Megha Agarwal, and Rajesh Kumar Mundotiya. 2020. Pits@ dravidian-codemix-fire2020: Traditional approach to noisy code-mixed sentiment analysis. In *FIRE (Working Notes)*, pages 541–547.

Kishorjit Nongmeikapam, Wahengbam Kumar, and Mithlesh Prasad Singh. 2017. Exploring an efficient handwritten Manipuri meetei-mayek character recognition using gradient feature extractor and cosine distance based multiclass k-nearest neighbor classifier. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 328–337, Kolkata, India. NLP Association of India.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1971–1982.

Ivan Tomek. 1976. Two modifications of cnn.

Atnafu Lambebo Tonja, Olumide Ebenezer Ojo, Mohammed Arif Khan, Abdul Gafar Manuel Meque, Olga Kolesnikova, Grigori Sidorov, and Alexander Gelbukh. 2022. CIC NLP at SMM4H 2022: a BERT-based approach for classification of social media forum posts. In *Proceedings of The Seventh Workshop on Social Media Mining for Health Applications, Workshop & Shared Task*, pages 58–61, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Mesay Gemeda Yigezu, Michael Melese Woldeyohannis, and Atnafu Lambebo Tonja. 2021. Multilingual neural machine translation for low resourced languages: Ometo-english. In *2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, pages 89–94.

Word Level Language Identification in Code-mixed Kannada-English Texts using Deep Learning Approach

Mesay Gemeda Yigezu[1], **Atnafu Lambebo Tonja**[2], **Olga Kolesnikova**[3], **Moein Shahiki Tash**[4], **Grigori Sidorov**[5], **Alexander Gelbukh**[6]

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Mexico City, Mexico
{mgemedak2022[1], [2]alambedot2022, [5]sidorov, [6]gelbukh}@cic.ipn.mx
{[3]kolesolga, [4] moein.tash}@gmail.com

Abstract

The goal of code-mixed language identification (LID) is to determine which language is spoken or written in a given segment of a speech, word, sentence, or document. Our task is to identify English, Kannada, and mixed language from the provided data. To train a model we used the CoLI-Kenglish dataset, which contains English, Kannada, and mixed-language words. In our work, we conducted several experiments in order to obtain the best performing model. Then, we implemented the best model by using Bidirectional Long Short Term Memory (Bi-LSTM), which outperformed the other trained models with an F1-score of 0.61%.

1 Introduction

Language identification is one of the most pernicious challenges in NLP. It is also a difficult task to handle bilingual and multilingual communication data. The prevalence of multilingualism on the internet, and code-mixed text data, has become a popular research topic in NLP. Several strategies have been explored over the years to assess and attempt to identify the document's languages and classify each text based on its language from some closed set of known languages. In today's bilingual or multilingual societies, many users regularly switch back and forth between two or more languages when typing and communicating, a process known as code-mixing or code-switching (Mandal and Singh, 2018). Although much effort has recently been directed toward this issue, the challenge of language tagging in the code-mixed scenario remains unresolved. The freedom of expression allows users to express and convey their thoughts in real-time all over the world, some people publishing content using more than one language which results in code-mixed text (Dowlagar and Mamidi, 2021; Andrew, 2021; Yigezu et al., 2021; Tonja et al., 2022). One of the problems related to this issue is translation, given a source text, the translation system fails to translate it into the targeted language due to the linguistic mixture (Smith and Thayasivam, 2019) if it does not include a module to identify the language in the original text.

In order to address the problem of word-level language identification, particularly in Kannada-English texts COLI-Kanglish shared a task provided for us. So, as part of this task, we looked at how different state-of-the-art techniques are used and came up with a model to find Kannada and English words in code-mixed text.

2 Related Work

Language identification is one of the oldest NLP problems (Beesley, 1988), especially in regards to spoken language (House and Neuburg, 1977), and code-switching was often considered a substandard use of language. In addition to that, in the recent past, a lot of work has been done in the field of code-mixed data analysis. In order to obtain and understand the state of the art, we have reviewed various related research, from those research works, we selected three papers which are more representative in our opinion.

Mandal and Singh (2018) put into practice multichannel neural networks incorporating CNN and LSTM for word-level language identification of code-mixed data. They combined this with a Bi-LSTM-CRF context capture module and obtained an accuracy of 93.28% and 93.32% evaluated on two test data sets respectively.

Das and Gambäck (2014) looked at chat message English Bengali and English-Hindi corpora to identify language borders at the word level. To determine the level of language blending in the corpora and define the effectiveness of a system designed to distinguish several languages, they proposed a code-mixing index. They primarily employed conventional methods such as character n-grams, dictionaries, and SVM classifiers.

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 29 - 33
December 15-18, 2022 ©2022 Association for Computational Linguistics

King and Abney (2013) investigated methods for word-level language identification in texts with multiple languages. They gathered and manually analyzed a corpus of over 250,000 words of bilingual (primarily non-parallel) content from the web to assess their methodologies. They experimented with different combinations of character unigrams, bigrams, trigrams, 4-grams, 5-grams, and the whole word using a logistic regression classifier.

3 Task description

Word level is the smallest unit of code-mixing. The code-mixed data is limited in resources, and the models that help to interpret them are still being developed (Dowlagar and Mamidi, 2021). These include identifying hate speech and fake speech, tagging parts of speech, shallow parsing, named entity recognition, etc. An improvement in these tasks can aid in the code-mixed dataset's syntactic and semantic analysis as well as the identification of code-mixed languages. Our task is to identify each code-mixed language, where we considered a word-level approach. It is a challenging task because we can not obtain huge data from various domain perspectives to train a model getting and better performance. The task of automatically identifying languages used in a given text is called language identification(LI). For many applications, LI serves as a preprocessing step. At the word level, LI may be thought of as a sequence labeling issue where each word in a sentence is assigned to either a mixed language or one of the languages in a specified set of languages. Despite a lot of work being done in LI, the problem of LI in the code-mixed scenario is still a long way from being resolved. Balouchzahi et al. (2022) Kannada is one of the Dravidian languages spoken in the Karnataka state in India. Karnataka residents can read, write, and speak Kannada, yet many have trouble using the language to send messages or make comments on social media.

4 Data description

While technological limitations like the keyboards of computers and smartphones are one reason, another may be the complexity of framing words with consonant conjuncts. As a result, the majority of individuals who post comments on social media do so using only Roman writing or a combination of Kannada and Roman letters. To address word level LI in code-mixed Kannada-English (Kn-En) texts, these texts are taken from Kannada YouTube video comments to construct Code-mixed Language Identification (CoLI-Kenglish) dataset (Hosahalli Lakshmaiah et al., 2022). In this task, the primary challenging activity is data collection, which is done by the organizer. we obtained data that contains 19, code-mixed data at the word level. The collected data corpus has two attributes, which are words and tags. For each word, a language identification tag was assigned. The tags were 'en-kn', 'en', 'kn', 'name', 'location' and 'other'.

Word	Tag
chennai	location
nandu	kn
soori	name
gida	kn
tailor	en
tamilan	other
kannadanu	en-kn

Table 1: Sample data

The 'en' and 'kn' were assigned to words that are present in the English language and the Kannada languages, respectively. The 'en-kn' is assigned to a word that contains both English and Kannada. The 'name' tag was assigned to any type of named entity.' Location' was assigned to a word that can refer to a place, and the rest of the words are assigned the 'other' tag.

Figure 1 depicts each tag percentage in our task. The tags are not balanced, as seen in the above figure 1, which could result in an inaccurate LI outcome indicating, high bias and low variance when a model is unable to capture the underlying pattern of each tags. It occurs when we try to build a linear model using a nonlinear one or when we have very few tags to build an appropriate model.

4.1 Training and Testing dataset

To build a word-level model, we used 14,847 words, and the rest of the data (4,585 words) were reserved for testing the trained model. All data we used in training and testing is tagged.

Figure 2 depicts the distribution of datasets mentioned above.

5 System description

We used Torch, a deep learning framework, to train and develop our model. Popular libraries for neural

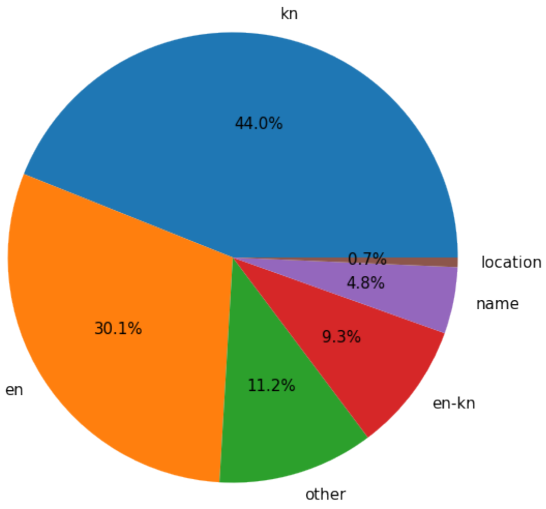


Figure 1: data-set size

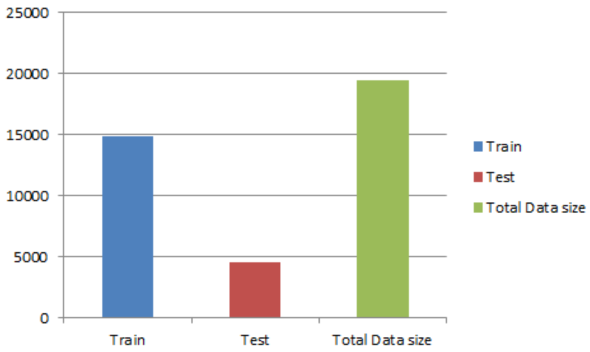


Figure 2: data-set size

networks and optimization are easy to use, they are flexible enough for creating complex neural network typologies. A few assistance functions ’Tensor.topk’ were created before we started training. The first step was to interpret the network’s output, which is the probability for each category, as we are aware of it. To obtain the index with the highest value and pass an input and a previous hidden state in order to perform a step of this network, The output (probability of each language) and the next hidden state was returned to us (which we kept for the next step). We used line tensors and slices, which could be further optimized by pre-computing batches of tensors. Finally, We generated a confusion matrix to determine how well the network performed on various tags for each language. A large number of samples were processed by the network using evaluate(), which is the same as the train() but without the backdrop, to obtain the confusion matrix.

5.1 Hyper parameter setting

We conducted many experiments to choose the parameters, and finally, the following parameters were defined for the Bi-LSTM model. **Dataset Ratio:-** 80% training and 20% evaluation split gave better results.

Batch Size:- We utilized a maximum batch size of 256, which is preferred in model training, to decrease the machine’s processing time and achieve good results.

Epochs:-In the experiments, the model was trained using epochs ranging from 10 to 100. During the training phase, we observed that if we utilized too few or too many epochs, there is a wide disparity between the training error and the model’s validation error. After several attempts, the model got optimal results after 30 epochs.

Optimization algorithms:-We used the Adaptive Moment Estimation (Adam) optimizer, which updates the model’s weights and optimizes its parameters.

Loss function:-We used nn.CrossEntropyLoss() criterion combines nn.LogSoftmax() and nn.NLLLoss() in one single class. It was useful during training .

6 Experiments and Results

In this task, we explored techniques for performing language identification at the word level in the code-mixed language. In order to train and build a better model we have conducted various techniques.

From the deep learning side, we built and trained a basic character-level RNN to identify words. Character level RNNs read words as a sequence of characters, producing predictions and hidden states at each step and feeding their most recent hidden state, to the preceding step. RNN was often used as a building block in more recent neural networks to identify languages. We implemented both basic unidirectional LSTM model and Bi-LSTM models for code-mixed language identification with and without attention.

The model started with an embedding layer, then two layers of Bi-LSTM, and finally, an attention. Following this attention layer was a dense layer with ReLU activation. Then our model identified itself with the help of a dense layer with softmax activation. Various experiments revealed that Bi-LSTM performed with greater accuracy and an F1-score of 0.61%compared with the rest of the RNN

Table 2: Experimental results

Techniques	Weighted			Macro		
	Precision	Recall	F1- score	Precision	Recall	F1-score
LSTM	0.84	0.83	0.83	0.60	0.56	0.56
Bi-LSTM	0.84	0.82	0.82	0.61	0.58	0.57
LSTM with attention	0.84	0.83	0.83	0.61	0.57	0.57
Bi-LSTM with attetion	0.85	0.83	0.84	0.66	0.60	0.61
Random Forest	0.84	0.83	0.82	0.62	0.54	0.55

models and other techniques. In addition to this experiment, we attempted to build a model using the random forest machine learning technique, which performed less efficiently in our scenario than the other techniques mentioned above. Finally, we advise researchers in the LI area to collect enough data for the entire perspective, increase number of features, and set aside time for training. Table 2 presents the results of our experiments, using macro-averaged and weighted-averaged scores.

As it can be seen in Table 2, our results show that the Bi-LSTM with Attention performed better on the supplied code-mixed language than the other RNN models. It is due to the presence of an attention mechanism in the model. The attention method finds each word in the given code-mixed languages, which helps the model perform better than the other models. Although it is better than the other models, the results obtained are not satisfactory. There are various reasons for this and one of them is the complex nature of the code-mixing language and the presence of sarcastic tags as we have discussed in section 4.

7 Conclusion

As shown in table 2, all models are quite close in terms of F1-score. Bi-LSTM, on the other hand, is the most accurate model to utilize for the job of word-level language detection in code-mixed texts. The weighted averages for the precision, recall and F-score for the task at hand is shown in table 2. A precision of 0.66, a recall of 0.60 and an F1-score of 0.61 is achieved by the method presented in this paper to identify Kannada and English languages. there the result shows thet Bi-LSTM with attention better perform for language identification problem. It allows you to examine a specific sequence both front to back and back to front. When input data is received, the LSTM structure learns how much of the prior network state to apply. Information can flow in both directions when the hidden state is used. The outputs of the two LSTMs are blended at each time step because the BiLSTM model removes the barriers of conventional RNNs, it gives promising result.

Acknowledgement

The work was done with partial support from the Mexican Government through the grant A1S-47854 of CONACYT, Mexico, grants 20220852, 20220859, and 20221627 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

Judith Jeyafreeda Andrew. 2021. Judithjeyafreedaandrew@ dravidianlangtech-eacl2021: offensive language detection for dravidian code-mixed youtube comments. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 169–174.

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*.

Kenneth R Beesley. 1988. Language identifier: A computer program for automatic natural-language identification of on-line text. In *Proceedings of the 29th annual conference of the American Translators Association*, volume 47, page 54.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text.

Suman Dowlagar and Radhika Mamidi. 2021. A survey of recent neural network models on code-mixed indian hate speech data. In *Forum for Information Retrieval Evaluation*, pages 67–74.

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*.

Arthur S House and Edward P Neuburg. 1977. Toward automatic identification of the language of an utterance. i. preliminary methodological considerations. *The Journal of the Acoustical Society of America*, 62(3):708–713.

Ben King and Steven Abney. 2013. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1110–1119.

Soumil Mandal and Anil Kumar Singh. 2018. Language identification in code-mixed data using multichannel neural networks and context capture. *arXiv preprint arXiv:1808.07118*.

Ian Smith and Uthayasanker Thayasivam. 2019. Language detection in sinhala-english code-mixed data. In *2019 International Conference on Asian Language Processing (IALP)*, pages 228–233. IEEE.

Atnafu Lambebo Tonja, Olga Kolesnikova, Muhammad Arif, Alexander Gelbukh, and Grigori Sidorov. 2022. Improving neural machine translation for low resource languages using mixed training: The case of ethiopian languages. In *Mexican International Conference on Artificial Intelligence*, pages 30–40. Springer.

Mesay Gemeda Yigezu, Michael Melese Woldeyohannis, and Atnafu Lambebo Tonja. 2021. Multilingual neural machine translation for low resourced languages: Ometo-english. In *2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, pages 89–94. IEEE.

BoNC: Bag of N-Characters Model for Word Level Language Identification

Shimaa Ismail[1] and **Mai K. Gallab**[2] and **Hamada Nayel**[2]
[1]Department of Information Systems
[2]Department of Computer Science
Faculty of Computers and Artificial Intelligence
Benha University, Egypt
{shimaa.mustafa, mai.gallab, hamada.ali}@fci.bu.edu.eg

Abstract

This paper describes the model submitted by NLP_BFCAI team for Kanglish shared task held at ICON 2022. The proposed model used a very simple approach based on the word representation. Simple machine learning classification algorithms, Random Forests, Support Vector Machines, Stochastic Gradient Descent and Multi-Layer Perceptron have been implemented. Our submission, RF, securely ranked fifth among all other submissions.

1 Introduction

Language Identification ("LI") is the process of identifying the natural language that a document or a portion of it is written in (Li et al., 2013). A human reader who is familiar with a language may easily recognize material written in that language. Therefore, the goal of LI research is to imitate the capacity of humans to identify these languages. The early attempts to solve this problem were made at the beginning of the century (Tomokiyo and Jones, 2001; Jarvis and Crossley, 2012). After that, there are several computer methods is being used without the assistance of a human using specifically created algorithms and indexing structures. Over the years, LI research has developed methods to recognize human languages. LI is applicable for all forms of information storage that incorporate language, whether digital or not, and applies to every modality of language, including voice, sign language, and handwritten text. However, we restrict the scope of this paper to LI of written material that has been digitally encoded.

The ability to identify the language for a written document has a wide range of uses such NLP tasks. It plays an important role in attracting users to a specific website that can provide relevant information for the user's native language (Kralisch and Mandl, 2006). In information retrieval and storage, the procedure of indexing documents in a multilingual collection according to the language they were written in is common. LI is required for document collections where the languages of the documents are unknown at the outset, such as for data retrieved from the World Wide Web (Jauhiainen et al., 2019). It is also suitable for machine translator applications by detecting the user's language without selecting it.

Language Identification is useful also in security. Forensic linguistics is one of the potential applications that use the LI which is considered the link between the legal system and linguistic stylistics (McMenamin, 2002). LI can be used as a methodology for authorship profiling to give proof of an author's linguistic background (Grant, 2007). Authorities and intelligence agencies may be able to learn more about threats and their perpetrators if they can extract more information from an anonymous SMS. Investigators can identify the author of anonymous literature with the use of hints about their languages. In these situations, LI can be used to discover the discriminant language cues in anonymous communication (Abbasi and Chen, 2005).

The study of language acquisition and teaching has received a lot of linguistic attention. The need for resources for language learning has increased because of the growing number of language learners, which has in turn fueled most of the language acquisition research over the past ten years.

The development of the Second Language Acquisition (SLA) theory may potentially benefit from the outcomes of an LI task. The new detection-based approach to transfer articulates the convergence of LI approaches and transfer research (Jarvis, 2010), which was first proposed by Tsur and Rappoport (Tsur and Rappoport, 2007). LI can be used to create teaching strategies, guidelines, and learner feedback that are tailored to each

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 34 - 37
December 15-18, 2022 ©2022 Association for Computational Linguistics

student's mother language. The models specific to each language can be used to create this customized evaluation. For instance, algorithms based on these models could give students feedback in automated writing evaluation systems that is considerably more targeted and concentrated (Rozovskaya and Roth, 2011).

A new word-representation model, Bag of N-Characters (BoNC), has been presented in this work. The proposed model is a derivation of the Bag of Words model (BoW) for characters. Different machine learning algorithms namely; Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), Random Forest (RF) and Multi-Layer Perceptron (MLP) have been implemented using BoNC model as a vectorization technique.

The rest of the paper is organized as follows: Section 2 describes the dataset; section 3 describes the system architecture. Experimental settings and results are given in section 4. Finally, the conclusion and future work are presented in section 5.

2 Dataset

The dataset, CoLI-Kenglish, has been distributed by task organizers to the participants (Hosahalli Lakshmaiah et al., 2022). It comprises a set of English and Kannada words written in Roman script. The words are grouped into the following set of classes $\{Kannada, English, Mixed-language, Name, Location, Other\}$

3 System Architecture

The general framework of proposed model consists of three main phases. The first phase is preprocessing where the raw words were prepared to further steps. The second phase is word representation and the third phase is model training. After model construction, the test set was fed to the model for model evaluation. The following are details of each phase.

3.1 Preprocessing

The preprocessing step consists of creating a vocabulary of characters $\mathcal{V}$. In this work, we set the threshold for the occurrence of the character to be considered as $k = 4$.

$$\mathcal{V} = \{ch \,|\, \textbf{number of occurrence of } ch >= 4\}$$

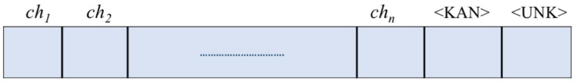


Figure 1: Word representation vector.

3.2 Word Representation

To represent the training samples, *words*, we used vector of length exceeds the number of characters in the set $\mathcal{V}$ by 2. The components of this vector represents the number of occurrence of the corresponding character as shown in figure 1. The last two components of the vector is reserved for the unknown characters, Kannada characters **<KAN>** and unknown characters **<UNK>**.

3.3 Model Construction

The training samples or words are now represented as vectors. Now, the current phase is model creation. Various machine learning classifiers, namely, support vector machines, random forests and multi-layer perceptron have been implemented.

3.3.1 Support Vector Machines

For text classification problems including a significant number of features and documents, as those in the current study. SVM is effective and demonstrated great promise in NLP applications such as dialect identification (Nayel et al., 2021b), rumors detection (Ashraf et al., 2022), sentiment detection (Nayel et al., 2021a), sarcasm detection (Nayel et al., 2021a) and gender biased detection (Elkazzaz et al., 2021).

SVM is a classification technique that generates statistical models that can differentiate between similar classes in the training data. By representing each example in the training data as a point in multidimensional space, it achieves this.

3.3.2 Random Forest (RF)

The random forest is a series of decision trees linked together by several bootstrap samples generated from the original data set. Based on the entropy (or Gini index) of a chosen subset of the features, the nodes are divided. The subsets that are generated using bootstrapping from the original data set, have the same size as the original data set size. Random forests can develop into quite sophisticated predictive models.

3.3.3 Multi-layer Perceptron (MLP)

MLP are adjuncts to feedforward neural networks. It is often used in supervised learning. MLP consists of three types of layers: input layer, output layer and hidden layer. Each layer consists of nodes. The output layer node represents the set of class labels present in the training data set. Learning process in MLP consists of adjusting perceptron weights to make the training data less in errors. This is traditionally done using a back-propagation algorithm that attempts to minimize the MSE.

3.4 Performance Evaluation

We calculated four evaluation metrics, Precision (P), Recall (R), and F1-score to measure the performance of our models. The macro f1-score is the official metric for the shared task (Balouchzahi et al., 2022).

4 Experiments and Results

For preprocessing phase, the threshold is set to be 4, $k = 4$. The vector length was 64, i.e the character vocabulary contains 64 characters. K-folds cross validation technique has been used for the development phase. The training set is divided into three folds, at the first run the first fold has been used as test set and other folds as the training set and so on. Table 1 shows the results of all classifiers for the 3-fold cross validation technique.

Algorithm	Accuracy
RF	64.89%
SGD	65.19%
SVM (Linear)	62.94%
MLP (h=10)	64.49%
MLP (h=20)	64.97%
MLP (h=40)	65.38%

Table 1: 3-fold cross validation accuracy for all classifiers.

Table 2 shows the result of our submission for the shared task for all classifiers. RF proved its superiority and achieved the best performance.

5 Conclusion and Future Work

In this paper, a simple framework for language identification has been introduced. A vectorization approach (BoNC) has been compared. It is clear from the results that RF outperforms all other classifiers. From this study, we can conclude that language identification of text is one of the challenging tasks.

In future work, pre-trained models could be used to improve the performance of classification. Transfer learning can be applied so that knowledge from one domain can be transferred to another domain.

References

A. Abbasi and H. Chen. 2005. Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20(5):67–75.

Nsrin Ashraf, Hamada Nayel, and Mohamed Taha. 2022. A comparative study of machine learning approaches for rumors detection in covid-19 tweets. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 384–387.

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*.

Fathy Elkazzaz, Fatma Sakr, Rasha Orban, and Hamada Nayel. 2021. BFCAI at ComMA@ICON 2021: Support vector machines for multilingual gender biased and communal language identification. In *Proceedings of the 18th International Conference on Natural Language Processing: Shared Task on Multilingual Gender Biased and Communal Language Identification*, pages 70–74, NIT Silchar. NLP Association of India (NLPAI).

Tim Grant. 2007. Quantifying evidence in forensic authorship analysis. *International Journal of Speech, Language & the Law*, 14(1).

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*.

Scott Jarvis. 2010. Comparison-based and detection-based approaches to transfer research. *EUROSLA Yearbook*, 10(1):169–192.

Scott Jarvis and Scott A. Crossley, editors. 2012. *Approaching Language Transfer through Text Classification*. Multilingual Matters, Bristol, Blue Ridge Summit.

	Weighted			Macro F1		
Algorithm	**P**	**R**	**F1-score**	**P**	**R**	**F1-score**
RF	0.73	0.73	0.72	0.52	0.41	0.43
SGD	0.74	0.73	0.72	0.51	0.43	0.41
SVM	0.73	0.73	0.72	0.42	0.36	0.36
MLP($h = 20$)	0.73	0.72	0.70	0.43	0.34	0.34
MLP($h = 10$)	0.72	0.72	0.70	0.42	0.34	0.34

Table 2: Results of our submissions on test set

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *J. Artif. Int. Res.*, 65(1):675–682.

A. Kralisch and T. Mandl. 2006. Barriers to information access across languages on the internet: Network and language effects. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 3, pages 54b–54b.

Haizhou Li, Bin Ma, and Kong Aik Lee. 2013. Spoken language recognition: From fundamentals to practice. *Proceedings of the IEEE*, 101(5):1136–1159.

Gerald R McMenamin. 2002. *Forensic linguistics: Advances in forensic stylistics*. CRC press.

Hamada Nayel, Eslam Amer, Aya Allam, and Hanya Abdallah. 2021a. Machine learning-based model for sentiment and sarcasm detection. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 386–389, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Hamada Nayel, Ahmed Hassan, Mahmoud Sobhi, and Ahmed El-Sawy. 2021b. Machine learning-based approach for Arabic dialect identification. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 287–290, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

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*, pages 924–933, Portland, Oregon, USA. Association for Computational Linguistics.

Laura Mayfield Tomokiyo and Rosie Jones. 2001. You're not from 'round here, are you? naive Bayes detection of non-native utterances. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Oren Tsur and Ari Rappoport. 2007. Using classifier features for studying the effect of native language on the choice of written second language words. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 9–16, Prague, Czech Republic. Association for Computational Linguistics.

Overview of CoLI-Kanglish: Word Level Language Identification in Code-mixed Kannada-English Texts at ICON 2022

F. Balouchzahi[1,*] **S. Butt**[1], **A. Hegde**[2], **N. Ashraf** [3],
H.L. Shashirekha[2], **G. Sidorov**[1], and **A. Gelbukh**[1]
[1]Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico
[2]Department of Computer Science, Mangalore University, Mangalore, India
[3]Dana-Farber Cancer Institute, Harvard Medical School, United States
Corresponding: *fbalouchzahi2021@cic.ipn.mx

Abstract

The task of Language Identification (LI) in text processing refers to automatically identifying the languages used in a text document. LI task is usually been studied at the document level and often in high-resource languages while giving less importance to low-resource languages. However, with the recent advancement in technologies, in a multilingual country like India, many low-resource language users post their comments using English and one or more language(s) in the form of code-mixed texts. Combination of Kannada and English is one such code-mixed text of mixing Kannada and English languages at various levels. To address the word level LI in code-mixed text, in CoLI-Kanglish shared task, we have focused on open-sourcing a Kannada-English code-mixed dataset for word level LI of Kannada, English and mixed-language words written in Roman script. The task includes classifying each word in the given text into one of six predefined categories, namely: Kannada (kn), English (en), Kannada-English (kn-en), Name (name), Location (location), and Other (other). Among the models submitted by all the participants, the best performing model obtained averaged-weighted and averaged-macro F1 scores of 0.86 and 0.62 respectively.

1 Introduction

South Asia is the most linguistically diverse region in the world that embodies more than 650 different languages[1] and India is a multilingual country having a rich heritage of languages in South Asia. Kannada is one of the Dravidian[2] languages as well as scheduled languages of India and the official and administrative language of Karnataka state with more than 40 million native Kannada speakers. A significant number of people in this region are comfortable using English in addition to their native/local/regional language for the day-to-day communication. These multilingual speakers preferably use multiple scripts and/or languages to post their comments/ideas/opinions on social media platforms, making code-mixing a default language on social media.

Code-mixing can be carried out at the paragraph, sentence or word level and even at subword level (Chakravarthi et al., 2020; Hegde et al., 2022a). People usually mix their native and/or local language with English and prefer to write the content mostly in Roman script rather than using the native script as most of the keyboard layouts of computers and keypads of smartphones have Roman alphabets by default (Balouchzahi et al.).

People who write Kannada find difficult to use Kannada script while posting comments/reviews on social media mainly because of the difficulty in keying the consonant conjuncts (*ottaksharas*) and consonants with the secondary forms of vowels (*gunitaskaras*) (Kittel, 1903), using Roman keyboards/keypads and hence prefer to use Roman script on social media (Balouchzahi et al., 2021b). The situation remains the same for most of the Indian languages as they have their own script.

Social media platforms have given their users the freedom of writing text very casually without following the grammar or syntax of any language. This has resulted in a huge volume of user-generated content which includes incomplete words and/or sentences, catchy phrases, user-defined short forms for words (e.g., 'gn8' for 'good night'), different slangs (e.g., meme, Gmeet), abbreviations ('OMG' for 'Oh my God'), recurrent characters ('soooooo sad' for 'so sad'), etc. The presence of these informal words in any text makes it difficult to understand the content (Shashirekha et al., 2022). Further, a code-mixed scenario where words of one language are transcribed with words of other languages as prefixes or suffixes creates a

[1] https://www.deccanherald.com/content/652273/intl-meet-south-asian-languages.html

[2] https://en.wikipedia.org/wiki/Kannada

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 38 - 45
December 15-18, 2022 ©2022 Association for Computational Linguistics

lot of problems to analyze the text, particularly due to conflicting phonetics.

The increasing number of social media users is increasing the user-generated content which makes it difficult to handle this text manually (Scotton, 1982). This demands the tools and techniques that can process the user-generated text automatically for various applications.

The preliminary step in handling code-mixed text for many of the Natural Language Processing (NLP) tasks like Machine Translation (Patel and Parikh, 2020), Parts-Of-Speech tagging (Dowlagar and Mamidi, 2021), Sentiment Analysis (Bansal et al., 2020; Balouchzahi et al., 2021c; Balouchzahi and Shashirekha, 2021), Emotion Analysis (Hegde et al., 2022b), Hate Speech and Offensive Language Identification (Balouchzahi et al., 2021a; Hegde et al., 2021), Hope Speech Detection (Gowda et al., 2022), Identification of Native Language (Nayel and Shashirekha, 2018, 2017), etc., is identifying the language of each word/phrase/sentence.

Several research works in LI tasks have been carried out focusing on high-resource languages like French-English, Spanish-English, and German-English. However, very little attention is given to the low-resource Indian languages. Furthermore, code-mixing is quite common in a multilingual country like India where many people are bilingual and English is considered as the official language along with the local/administrative language. Hence, in India, code-mixing is mostly observed between any Indian language and English in social media text (Balouchzahi and Shashirekha, 2020).

The rapidly increasing code-mixed content on social media in Indian languages in general and Kannada-English in particular requires efficient methods to perform LI at word level.

2 Literature Review

Recent decades have witnessed the immense interest of researchers in code-mixed text specifically for low-resource and under-resource languages and few LI works have also been carried out as a part of handling such code-mixed text. Word level LI is modeled as a typical supervised learning problem and various Machine Learning (ML) and Deep Learning (DL) algorithms are experimented for the same. Some of the relevant works are described below:

(Chaitanya et al., 2018) developed learning models for word level LI of Hindi-English code-mixed data using feature vectors generated by the Continuous Bag of Words (CBOW) and Skipgram models. They experimented with various ML models including: Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Gaussian Naive Bayes (GNB), k-Nearest Neighbor (kNN), and Adaptive Boosting (Adaboost), on the dataset consisting of 7,210 words selected from the corpus prepared by (Jamatia and Das, 2016). Among all the models, SVM classifiers obtained the highest accuracies of 67.33% and 67.34% using CBOW and Skipgram models respectively. A word level LI in code-mixed Telugu-English text proposed by (Gundapu and Mamidi, 2020), tokenized 1,987 Telugu-English code-mixed sentences obtained from the International Conference on Natural Language Processing (ICON) 2015 shared task dataset[3] and manually tagged the tokenized words with Parts-Of-Speech (POS) and LI tags. By using previous, current and next words and their POS tags, length of the word, and character n-grams in the range n = (1, 3) as features, they trained Conditional Random Field (CRF) classifier to perform word level LI and obtained an accuracy of 91.28%.

(Mandal and Singh, 2018) proposed a multichannel Neural Network (NN) model for LI of code-mixed Hindi-English and Bengali-English text using contextual information. They selected 6,000 instances from the dataset developed by (Patra et al., 2018) and Mandal et al. (2018) for Hindi-English and Bengali-English respectively and implemented multichannel neural associations by combining Convolutional Neural Network (CNN) and Long short-term memory (LSTM) models coupled with BiLSTM-CRF for word level LI. Their proposed models obtained accuracies of 93.32% and 93.28% for Hindi-English and Bengali-English data respectively. (Thara and Poornachandran, 2021) created a dataset for word level LI in code-mixed English-Malayalam text and implemented transformer-based models for LI. The authors extracted 50K code-mixed English-Malayalam comments from YouTube and tokenized them to obtain 7,75,430 words. These words are then annotated with the language to which they belong to using an unsupervised approach. Transformer-based multilingual Bidirectional Encoder Representations from Transformers (mBERT), Cross-lingual

[3] https://ltrc.iiit.ac.in/icon2015/

Language Model for Robustly Optimized BERT (XLM-RoBERTa), CamemBERT, Distilled version of BERT (DistilBERT), and Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) models, are fine-tuned to perform LI. Among all the models, fine-tuned ELECTRA model performed best with an F1 score of 0.9933.

Word and Character embedding-based learning models presented by (Veena et al., 2018) for LI of code-mixed Hindi-English text are experimented on ICON 2016 shared task dataset (Jamatia and Das, 2016) that consists of 772, 1,096, and 763 sentences from Facebook, Twitter, and WhatsApp respectively. By collecting additional code-mixed Hindi-English text from other resources, word and character ngrams are used to train three skip-gram models with n = 1, 3, 5 which in turn is used to train the SVM models for LI. Compared to various SVM models trained on word-based and character-based embeddings, SVM classifier trained on character-based 5-gram embeddings obtained better accuracy.

Even though few research works are carried out in low-resource Indian languages like Kannada, Tamil, Telugu, etc., no works have been reported on word level LI in code-mixed Kannada-English text. This provides scope for research at word level LI in code-mixed Kannada-English text.

3 Task Description

The task of automatically identifying languages used in a given text is called LI and it is a preprocessing step for many applications. LI at the word level can be viewed as a sequence labeling problem where each and every word in a sentence is tagged with one of the languages in the predefined set of languages. Despite a lot of work being done in LI, the problem of LI in the code-mixed scenario is still a long way from being illuminated (Mandal and Singh, 2018).

To address word level LI in code-mixed Kannada-English texts, these texts are extracted from Kannada video comments in YouTube to construct CoLI-Kenglish (Shashirekha et al., 2022) dataset.

4 Dataset

Comments for Kannada videos in YouTube are scraped using the youtube-comment-downloader[4] and are used to build CoLI-Kenglish dataset (Shashirekha et al., 2022). The scraped texts contain around 1,00,000 comments from 373 Kannada YouTube videos. Preprocessing involves the removal of duplicate comments and comments written only in Kannada script. After preprocessing, the total number of comments amounts to 72,815. The nature of comments are generally in one of the following forms:

- Only in Kannada
- Only in English
- Combination of Kannada and English
- Other languages e.g., Hindi, Telugu and Tamil

A random sample of around 10% of the text is annotated by two native Kannada speakers to generate CoLI-Kenglish dataset and the rest of raw text is released as additional Kannada-English code-mixed resource.

The annotated CoLI-Kenglish dataset contains 19,432 unique words extracted from nearly 7,000 sentences that are categorized into 6 classes, namely: ‘Kannada’, ‘English’, ‘Mixed-language’, ‘Name’, ‘Location’ and ‘Other’. While ‘Kannada’ and ‘English’ classes represent Kannada and English words respectively, ‘Mixed-language’ class represents words created using a combination of Kannada and English in any order. ‘Name’ class represents the names of persons and ‘Location’ class the names of locations or places. Any other words are represented as an ‘Other’ class. The words described by ‘Mixed-language’ pose a real challenge to LI task as these words are framed by various combinations of English/Kannada words and Kannada/English affixes (prefixes and suffices). The beauty as well as the complexity of these mixed-language words lies in the word pattern created by an individual posting comments on social media. Description of the class labels and their samples along with the English translation are presented in Table 1 and the statistics of CoLI-Kenglish dataset in terms of Train and Test set are shown in Table 2. The statistics of the CoLI-Kenglish dataset illustrates that the dataset is imbalanced.

[4] https://github.com/egbertbouman/youtube-comment-downloader

Category	Tag	Description	Samples
Kannada	kn	Kannada words written in Roman script	kopista (one who get angry soon), baruthe (will come), barbeku (must come)
English	en	Pure English words	small, need, take, important
Mixed-language	kn-en	Combination of Kannada and English words in Roman script	coolagiru (cool + agiru, be cool), leaderge (leader + ge, to a leader), homealli (home + alli, inside home)
Name	name	Words that indicate name of person (including Indian names)	Madhuswamy, Hemavati, Swamy
Location	location	Words that indicate locations	Karnataka, Tumkur, Bangalore
Other	other	Words not belonging to any of the above categories and words of other languages	Znjdjfjbj – not a word kannada words in kannada script hindi words in Devanagari script hindi words in Roman script tamil words in Tamil script

Table 1: Description of the classes and their samples in CoLI-Kenglish dataset

Tag	Train set	Test set
kn	6,526	2,194
en	4,469	1,812
kn-en	1,379	93
name	708	354
location	102	31
other	1,663	100
Total	14,847	7,241

Table 2: Statistics of Train and Test set

5 Evaluation Metrics

In the case of an imbalanced dataset, categories with a larger number of samples affect the averaged-weighted scores and could be high always. Therefore, reporting only weighted scores could provide misleading information about models' performance. Hence, inspired by (Balouchzahi et al., 2022), code-mixed LI models for imbalanced CoLI-Kenglish dataset are evaluated using macro-averaged and weighted-averaged F1 scores.

6 Baselines

CoLI-ngrams - the best performing model proposed by (Shashirekha et al., 2022) employ a feature engineering module that generates a feature set of prefixes and suffixes of length 1, 2 and 3 along with char n-grams (n = 2, 3, 5) from words, and Byte-Pair Encoding (BPE) embeddings of sub-word n-grams (n = 1, 2, 3). The extracted features are vectorized using TfidfVectorizer[5] to train Linear SVM (LSVM), Multi-layer Perceptron (MLP), and Logistic Regression (LR) classifiers. These three models are used as baselines in this shared task. All the models are trained with default parameters.

7 Overview of the Submitted Approaches

Thirty different runs are submitted by eight different teams for the Kanglish 2022 shared task and eventually six teams submitted their working notes. Figure 1 refers to the different learning approaches used by the participants in this shared task to submit the runs. The findings indicate that, while 54% of the participants experimented different transformers, 27% used traditional ML models and the remaining used DL models. Figure 2 shows that about 46% of run submissions are made by employing pretrained Language Model (LM) or pretrained embeddings and 27% did not use any pretrained models for the task.

Team Tiya1012 presented a transformer-based model by fine-tuning DistilBERT-based-cased model on the CoLI-Kenglish dataset and obtained 0.62 averaged-macro F1 score and ranked first in the competition.

Team Abyssinia experimented different LM models, namely: BERT, mBERT, XLM-R and RoBERTa from HuggingFace with a LSTM architecture. Among all the LM models, both mBERT and XLM-R with an averaged-macro F1 score of 0.61 outperformed the rest of the models and also ranked second in the shared task.

Team PDNJK also explored several transformer-based models for the task of LI in code-mixed Kannada-English words and their best performing

[5] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

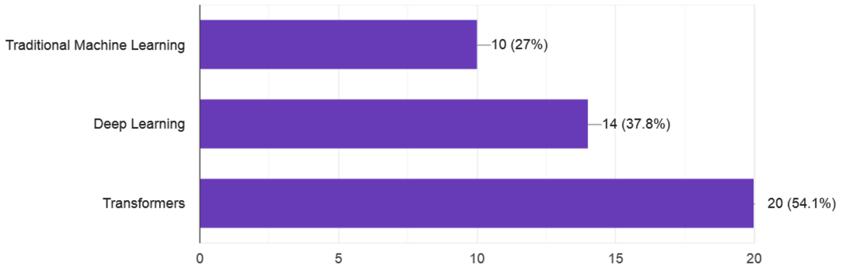


Figure 1: Learning approaches used by participants

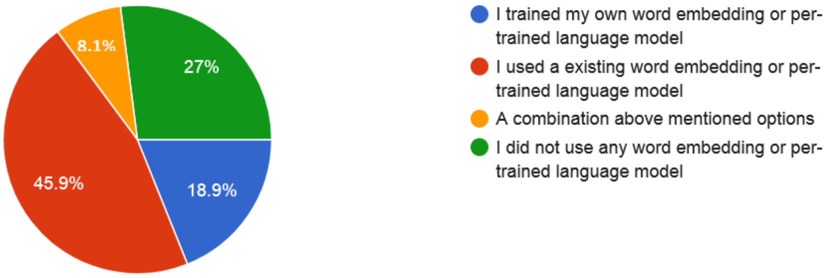


Figure 2: Pretrained models used by participants

model using BERT scored an averaged-macro F1 score of 0.57 and ranked fourth in the shared task.

Team Habesha trained character-level LSTM and BiLSTM models with attention that reads the text as a sequence of characters. The proposed BiLSTM model outperformed the LSTM model and obtained an averaged-macro F1 score of 0.61 and ranked second in the shared task.

Team Lidoma explored character n-grams to generate character TF-IDF to train traditional ML classifiers. Among all the classifiers they explored, the highest performance of an averaged-macro F1 score of 0.58 was reported with a simple kNN classifier. Similarly, Bag-of-Characters were turned into character vectors by **Team NLP_BFCAI**. They introduced a character representation called Bag-of-n-Characters model which has very similar structure to character n-grams and experimented several traditional ML algorithms. Eventually, the RF model on the proposed features obtained the highest averaged-macro F1 of 0.43.

8 Results and Discussion

The best results obtained for each team among all the predictions submitted by them are presented in Table 3 along with the results of the baseline models. Comparison of the results of the participating teams with that of the baseline models shows a slight improvement on F1-score for the first three best performing teams. The best averaged-macro F1 score of 0.62 shows the difficulty of the shared task. Further, our baselines utilizing n-grams generated from BPEmb sub-words, characters and affixes had a better performance of models that experimented only character n-grams.

Other findings indicate that, all teams who employed NN and transformer models outperformed the baselines and other traditional ML classifiers. In general, the higher weighted scores are the results of successful predictions for pure English and Kannada words and the difficulty on identifying mixed-language words and less frequent entities resulted in less performance for macro scores. Most of the teams relied on multilingual transformers or only character n-grams for solving the problem of LI in code-mixed text. This reveals that the participants have only a shallow understanding of code-mixed texts. No method was used by the participants that could directly target the issue of code-mixed texts except the multilingual transformers that partially handled the task.

9 Conclusion

The task of LI is a primary step for many NLP tasks that are usually overlooked for low-resource languages. However, the recent advancement in technologies caused a rapid increase in the volume of texts in low-resource languages. These texts on so-

Rank	Team name	Weighted			Macro		
		Precision	Recall	F1-score	Precision	Recall	F1-score
1	Tiya1012	0.87	0.85	0.86	0.67	0.61	0.62
2	Abyssinia	0.85	0.84	0.84	0.62	0.62	0.61
2	Habesha	0.85	0.83	0.84	0.66	0.6	0.61
-	LSVM-Baseline	0.84	0.84	0.83	0.67	0.57	0.59
3	Lidoma	0.83	0.83	0.83	0.64	0.56	0.58
4	PDNJK	0.86	0.85	0.86	0.58	0.58	0.57
-	MLP-Baseline	0.84	0.81	0.82	0.60	0.60	0.57
-	LR-Baseline	0.84	0.84	0.83	0.69	0.53	0.56
5	NLP_BFCAI	0.73	0.73	0.72	0.52	0.41	0.43
6	iREL	0.68	0.62	0.64	0.38	0.45	0.39
7	JUNLP	0.69	0.67	0.67	0.33	0.34	0.3
8	PresiUniv	0.57	0.59	0.53	0.22	0.22	0.2

Table 3: Participating team's best run score in the shared task

cial media are often a mixture of low-resource language with English resulting in code-mixed texts. In the code-mixed scenario, a sentence alone can have multiple languages at word level. Hence, the aim of Kanglish 2022 shared task was to promote word level LI for Kannada-English code-mixed texts. Initially, thirteen teams registered for the task and eventually more than thirty different runs were submitted by eight different teams. The majority of teams explored different NN models including transformers for the task. A fine-tuned DistilBERT model outperformed the rest of the models with averaged-weighted and averaged-macro F1 scores of 0.86 and 0.62 respectively.

The observation of performances of different models in the shared task reveals the difficulty of the LI task in code-mixed text. These difficulties are mainly due to the nature of code-mixed texts that do not follow the rules of and grammar of any language. This task aims to attract the attention of researchers for word level LI of different language pairs in code-mixed text. In future work, we would like to include more mixed-language words into CoLI-Kenglish dataset and also extend the corpus to different Dravidian languages including Tamil, Malayalam, etc.

Acknowledgements

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20220852 and 20220859 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

Fazlourrahman Balouchzahi, Aparna B K, and H L Shashirekha. 2021a. 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, Kyiv. Association for Computational Linguistics.

Fazlourrahman Balouchzahi, Aparna B K, and H L Shashirekha. 2021b. MUCS@LT-EDI-EACL2021:CoHope-Hope Speech Detection for Equality, Diversity, and Inclusion in Code-Mixed Texts. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 180–187, Kyiv. Association for Computational Linguistics.

Fazlourrahman Balouchzahi and H L Shashirekha. 2021. LA-SACo: A Study of Learning Approaches for Sentiments Analysis in Code-Mixing Texts. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 109–118, Kyiv. Association for Computational Linguistics.

Fazlourrahman Balouchzahi and HL Shashirekha. 2020. MUCS@ Dravidian-CodeMix-FIRE2020: SACO-Sentiments Analysis for CodeMix Text. In *FIRE (Working Notes)*, pages 495–502.

Fazlourrahman Balouchzahi, Hosahalli Lakshmaiah Shashirekha, and Grigori Sidorov. 2021c. CoSaD-

Code-Mixed Sentiments Analysis for Dravidian Languages. In *CEUR Workshop Proceedings*, pages 887–898.

Fazlourrahman Balouchzahi, Hosahalli Lakshmaiah Shashirekha, Grigori Sidorov, and Alexander Gelbukh. A Comparative Study of Syllables and Character Level N-grams for Dravidian Multi-script and Code-mixed Offensive Language Identification. In *Journal of Intelligent & Fuzzy Systems*, Preprint, pages 1–11. IOS Press.

Fazlourrahman Balouchzahi, Grigori Sidorov, and Alexander Gelbukh. 2022. PolyHope: Dataset Creation for a Two-Level Hope Speech Detection Task from Tweets. In *arXiv preprint arXiv:2210.14136*.

Neetika Bansal, Vishal Goyal, and Simpel Rani. 2020. Experimenting Language Identification for Sentiment Analysis of English Punjabi Code Mixed Social Media Text. In *International Journal of E-Adoption (IJEA)*, pages 52–62. IGI Global.

Inumella Chaitanya, Indeevar Madapakula, Subham Kumar Gupta, and S Thara. 2018. Word Level Language Identification in Code-mixed Data using Word Embedding Methods for Indian Languages. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1137–1141. IEEE.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John P McCrae. 2020. Corpus Creation for Sentiment Analysis in Code-Mixed Tamil-English Text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210.

Suman Dowlagar and Radhika Mamidi. 2021. A Pretrained Transformer and CNN Model with Joint Language ID and Part-of-Speech Tagging for Code-mixed Social-media Text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 367–374.

Anusha Gowda, Fazlourrahman Balouchzahi, Hosahalli Shashirekha, and Grigori Sidorov. 2022. MUCIC@LT-EDI-ACL2022: Hope Speech Detection using Data Re-Sampling and 1D Conv-LSTM. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 161–166, Dublin, Ireland. Association for Computational Linguistics.

Sunil Gundapu and Radhika Mamidi. 2020. Word Level Language Identification in English Telugu Code Mixed Data. In *arXiv preprint arXiv:2010.04482*.

Asha Hegde, Mudoor Devadas Anusha, Sharal Coelho, Hosahalli Lakshmaiah Shashirekha, and Bharathi Raja Chakravarthi. 2022a. Corpus Creation for Sentiment Analysis in Code-Mixed Tulu Text. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 33–40.

Asha Hegde, Mudoor Devadas Anusha, and Hosahalli Lakshmaiah Shashirekha. 2021. Ensemble based machine learning models for hate speech and offensive content identification. In *Forum for Information Retrieval Evaluation (Working Notes)(FIRE), CEUR-WS. org*.

Asha Hegde, Sharal Coelho, and Hosahalli Shashirekha. 2022b. MUCS@DravidianLangTech@ACL2022: Ensemble of Logistic Regression Penalties to Identify Emotions in Tamil Text. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 145–150, Dublin, Ireland. Association for Computational Linguistics.

Anupam Jamatia and Amitava Das. 2016. Task Report: Tool Contest on POS Tagging for Code-mixed Indian Social Media (Facebook, Twitter, and WhatsApp) text@ ICON 2016. In *Proceedings of ICON*.

Ferdinand Kittel. 1903. A Grammar of the Kannaḍa Language in English: Comprising the Three Dialects of the Language (Ancient, Mediæval and Modern). Basel Mission Book and Tract Depository.

Soumil Mandal, Sainik Kumar Mahata, and Dipankar Das. 2018. Preparing Bengali-English Code-mixed Corpus for Sentiment Analysis of Indian Languages. In *arXiv preprint arXiv:1803.04000*.

Soumil Mandal and Anil Kumar Singh. 2018. Language Identification in Code-Mixed Data using Multichannel Neural Networks and Context Capture. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 116–120. Association for Computational Linguistics.

Hamada A Nayel and HL Shashirekha. 2017. Mangalore-University@ INLI-FIRE-2017: Indian Native Language Identification using Support Vector Machines and Ensemble Approach. In *FIRE (Working Notes)*, pages 106–109.

Hamada A Nayel and HL Shashirekha. 2018. Mangalore University INLI@ FIRE2018: Artificial Neural Network and Ensemble based Models for INLI. In *FIRE (Working Notes)*, pages 110–118.

Devshree Patel and Ratnam Parikh. 2020. Language Identification and Translation of English and Gujarati Code-mixed Data. In *2020 International Conference on emerging trends in information technology and engineering (ic-ETITE)*, pages 1–4. IEEE.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment Analysis of Code-mixed Indian Languages: An Overview of Sail_Code-mixed Shared Task@ ICON-2017. In *arXiv preprint arXiv:1803.06745*.

Carol Myers Scotton. 1982. The Possibility of Code-Switching: Motivation for Maintaining Multilingualism. In *Anthropological linguistics*, pages 432–444.

Hosahalli Lakshmaiah Shashirekha, Balouchzahi Fazlourrahman, Mudoor Devadas Anusha, and Sidorov Grigori. 2022. CoLI-Machine Learning Approaches for Code-mixed Language Identification at Word Level in Kannada-English Texts. In *Special Issue of Acta Polytechnica*.

S Thara and Prabaharan Poornachandran. 2021. Transformer Based Language Identification for Malayalam-English Code-Mixed Text. In *IEEE Access*, pages 118837–118850. IEEE.

PV Veena, M Anand Kumar, and KP Soman. 2018. Character Embedding for Language Identification in Hindi-English Code-mixed Social Media Text. In *Computación y Sistemas*, pages 65–74. Instituto Politécnico Nacional, Centro de Investigación en Computación.

Author Index