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

Atnafu Lambebo Tonja¹, Mesay Gemeda Yigezu², Olga Kolesnikova³, Moein Shahiki Tash¹, Grigori Sidorov⁵, Alexander Gelbukh⁶

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Mexico City, Mexico
{¹alambedot2022, ²mgemedak2022, ³sidorov, ⁵gelbukh}@cic.ipn.mx
{³kolesolga, ⁴moein.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-Kenglish 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 Transformer based translation (Tonja et al., 2022b; Yigezu et al., 2021; Yigezu et al., 2021), abusive comment detection (Balouchzahi et al., 2022b), fake news detection (Arif et al., 2022; Truića 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-
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 CoLiKenglish 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. Figure 1 shows some samples from the dataset used for training.

![Figure 1: Training samples](image)

3.1 Dataset Statistics

Figures 2 and 3 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.

4 Methodology

This section presents a description of the data pre-processing, methodology, and models used in this 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.
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 4 shows the experimental architecture of our Transformer-based model for word level language identification in code-mixed Kannada-English texts. As shown in Figure 4, 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.
- **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 1 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 5. Figure 5 shows our proposed model summary for word-level language identification in code-mixed Kannada-English texts.

5.2 Results

Table 2 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 3. As shown in Table 3 our model ranked second in overall results among all participant teams.

\(^1\)https://colab.research.google.com/
\(^2\)https://scikit-learn.org/stable/
\(^3\)https://keras.io/
\(^4\)https://huggingface.co/
### Table 1: Transformers used in this paper and their parameters

<table>
<thead>
<tr>
<th>Model</th>
<th>Transformer blocks</th>
<th>Hidden layer size</th>
<th>Self-attention heads</th>
<th>#Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>bert-base-uncased</td>
<td>12</td>
<td>768</td>
<td>12</td>
<td>110M</td>
</tr>
<tr>
<td>bert-base-multilingual-uncased</td>
<td>12</td>
<td>768</td>
<td>12</td>
<td>110M</td>
</tr>
<tr>
<td>xlm-roberta-large</td>
<td>24</td>
<td>1024</td>
<td>16</td>
<td>355M</td>
</tr>
<tr>
<td>roberta-base</td>
<td>12</td>
<td>768</td>
<td>12</td>
<td>110M</td>
</tr>
</tbody>
</table>

### Table 2: Performance of our models on the test set (official results)

<table>
<thead>
<tr>
<th>Model</th>
<th>Weighted Score</th>
<th>Macro score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>bert-base-multilingual-uncased</td>
<td>0.84</td>
<td>0.85</td>
</tr>
<tr>
<td>xlm-roberta-large</td>
<td>0.83</td>
<td>0.8</td>
</tr>
<tr>
<td>roberta-base</td>
<td>0.85</td>
<td>0.84</td>
</tr>
</tbody>
</table>

### Table 3: Official rank of top 3 teams

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team name</th>
<th>Weighted Score</th>
<th>Macro score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>nyla1012</td>
<td>0.87</td>
<td>0.85</td>
</tr>
<tr>
<td>2</td>
<td>Our team</td>
<td>0.85</td>
<td>0.84</td>
</tr>
<tr>
<td>3</td>
<td>Habesha</td>
<td>0.85</td>
<td>0.83</td>
</tr>
<tr>
<td>3</td>
<td>lidoma</td>
<td>0.83</td>
<td>0.83</td>
</tr>
</tbody>
</table>

### Figure 7: Training and validation accuracy of BERT based approach

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.

### References


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.
