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

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

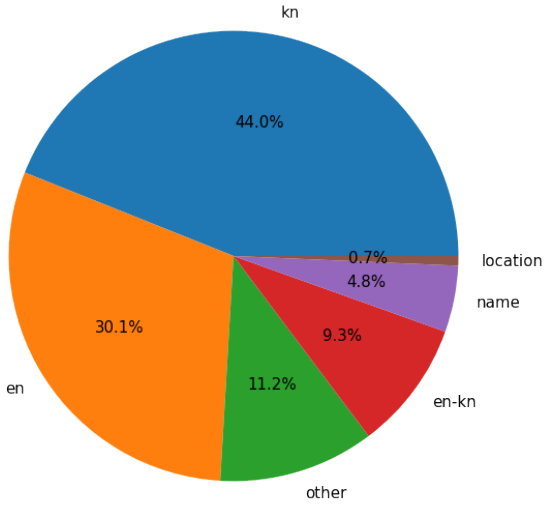

Figure 1: data-set size

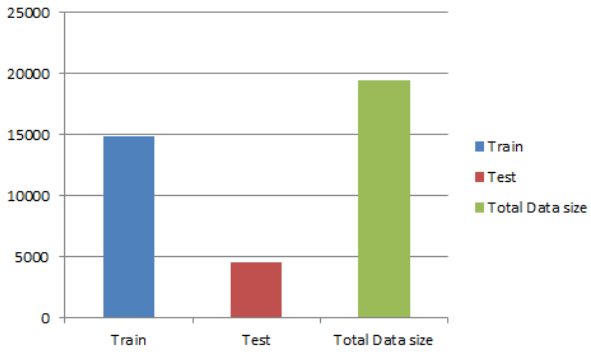

Figure 2: data-set size

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.