Corpus Creation and Language Identification in Low-Resource Code-Mixed Telugu-English Text

by

Anudeep Chaluvadi, Siva Subrahamanyam, Radhika Mamidi

in

RANLP

Report No: IIIT/TR/2021/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
September 2021

Corpus Creation and Language Identification in Low-Resource Code-Mixed Telugu-English Text

Subrahamanyam Varma, Anudeep Chaluvadi, Radhika Mamidi
Language Technologies Research Centre
International Institute of Information Technology
Hyderabad, Telangana, India
siva.subrahamanyam@research.iiit.ac.in
anudeep.chaluvadi@students.iiit.ac.in
radhika.mamidi@iiit.ac.in

Abstract

Code-Mixing (CM) is a common phenomenon in multilingual societies. CM plays a significant role in technology and medical fields where terminologies in the native language are not available or known. Language Identification (LID) of the CM data will help solve NLP tasks such as Spell Checking, Named Entity Recognition, Parts-Of-Speech tagging, and Semantic Parsing. In the current era of machine learning, a common problem to the above-mentioned tasks is the availability of Learning data to train models. In this paper, we introduce two Telugu-English CM manually annotated datasets (Twitter dataset and Blog dataset). The Twitter dataset contains more romanization variability and misspelled words than the blog dataset. We compare across various classification models and perform extensive bench-marking using both Classical and Deep Learning Models for LID compared to existing models. We propose two architectures for language classification (Telugu and English) in CM data: (1) Word Level Classification (2) Sentence Level word-by-word Classification and compare these approaches presenting two strong baselines for LID on these datasets.

1 Introduction

Language is one of the significant aspects which makes humans different from other species. It is not a fixed entity, and it has evolved with time and will continue to do so. As a part of such an evolutionary process, we are at the stage of Code-Mixing where people communicate by mixing linguistic units such as phrases, words, and morphemes of one language embedded within an utterance of another language (Sankoff and Poplack, 1981), (Poplack, 1980).

India has 22 officially recognized languages[1] and many dialects. For a land with such linguistic diversity, bilingualism and multilingualism is a prevalent trait. Telugu belongs to the family of Dravidian languages. It is primarily spoken in Southern India and is also the third most spoken language in India. Telugu is the mother language of a large population native of Andhra Pradesh and Telangana states. English is a primary mode of teaching for most of the population across the globe. With such an influence of a language, people tend to use a mix of both English and their native language in an informal conversation (both speech and text).

CM is classified into two types: Intra-Sentential and Inter-Sentential Code-Mixing (Zirker, 2007). Intra-Sentential Code Mixing refers to the use of multiple languages in a single sentence. Inter-Sentential Code-Mixing is when the language switching is done at the end of the sentence. In this paper, we will focus on Intra-Sentential Code Mixing.

In a multilingual setting, most of the conversations that happen informally are CM. CM is also used extensively in Social Media platforms, blogs, and forums as posts, chats, and comments. The processing of CM text poses an exciting and challenging problem to the linguistic community. This is because of the added complexities in the traditional processing tasks such as Spell Checking, Named Entity Recognition (NER), Parts-Of-Speech (POS), Natural Language Generation (NLG) and Machine Translation (MT) due to the unavailability of prior information about the language at any point of time. Intra-Sentential LID is the task of identifying the languages

[1] Eighth Schedule to the Constitution of India

of each word. Researchers have made significant progress in the LID module in the automated processing of CM text. However, in low-resourced agglutinative languages like Telugu, LID is still a challenging task (Parupalli et al., 2018).

In this paper, we introduce two datasets and propose various pipelines to tackle Intra-Sentential language identification problem in CM data using deep learning models. This example sentence illustrates the CM addressed in this paper:

Example: Elen/NE everyday/EN school/EN ki/TE buslo/TE velthundi/TE ./UNIV (Translation: Elen goes to school by bus everyday)

The words followed by /NE, /TE, /EN, and /UNIV correspond to Named Entity, Telugu, English, and Universal tags. In the above example, some words exhibit morpheme level Code-Mixing, like in "buslo" : "bus" (English word) + "lo" (plural morpheme in Telugu). We also consider the clitiques like "supere": "super" (English root word) + "e" (clitique) as code mixed.

The **key contributions** of this paper are the following:

- We open-sourced[2] two datasets of low-resourced CM English-Telugu data from popular global social media sites like Twitter and local blogging sites like Chaibisket.com, and Wirally.com.
- We propose extensive benchmarking with both Classical and Deep Learning Models for LID in CM data and infer that BiLSTM + CRF, BiLSTM + LSTM have higher classification metrics (overall and per-class) as compared with other models.
- We analyze the impact of contextual information of the word in a sentence for this task.

The rest of this paper is organised into 6 sections. In section 2, we discuss the related work. Section 3 elaborates on challenges while working with CM data, followed by the dataset and its annotation in section 4. Section 5 describes the approaches for LID in CM data. Section 6 reports the results of the proposed approaches. Finally, in section 7, we discuss the conclusions and future work.

[2] https://github.com/ksubbu199/cmtet-lid

2 Related Work

A significant amount of work has been done recently in the field of code-mixed data, especially in the area of LID. Kachru (1978) discussed the syntax and structure of multilingual language organization and the role of language dependence in linguistic convergence of CM from an Indian perspective.

To create a word-level language identifier, King and Abney (2013) used weakly semi-supervised methods. According to Noor Al-Qaysi (2017), code-switching is very popular on social networking sites such as Facebook, Twitter, and WhatsApp and 86.40 percent of students use code-switching on social networks, while 81 percent of educators do so.

Yogarshi Vyas and Choudhury (2014) used logistic regression and a module that calculates code-switching probability. Das and Gamb (2014) merged two classifiers into an ensemble model for Hindi-English CM LID using multiple features such as word sense, dictionary, n-gram, and edit distance. The first classifier takes changed edit distance, word frequency, and character n-grams as features. The first classifier's output and the POS tag of the neighboring words are given as features to the second classifier to predict the final label.

Sharma (2007) used the shallow parsing pipeline to perform successful text analysis in Hindi-English CM social media data. For the task of LID, most of the experiments depend on dictionaries, supervised classification models, and Markov models.

The basic distinguishing features such as specific character combinations, repeated or unique words, diacritics, or typical n-grams are used in the simplest LID methods (Dunning, 1994; Clive Souter and Johnson, 1994; Ciprian-Octavian Truic˘a and AlexandruBoicea, 2015).

Some LID methods model sequences of words, characters, or bytes as model complexity increases. Some approaches concentrate on modeling the frequency of n-grams, such as character n-gram frequency (Bashir Elhaj Ahmed and C.Tappert, 2004; Clive Souter and Johnson, 1994). These methods outper-

form techniques that rely on one-of-a-kind terms.

Kalita and Saharia (2018); Veena et al. (2018) used Support Vector Machines(SVM) with linear kernel to perform LID. Mave et al. (2018) equipped CRF based approach for LID on CM Hindi-English and Spanish-English text.

In (Gundapu and Mamidi, 2018), efforts have been made to propose an accurate model to address the problem of classification in CM data. However, the recent advancement in deep learning models have led to better performance in various NLP tasks. We thus leverage these deep models for classification involving low resourced language (Telugu) in CM data with the help of larger corpus which have been sourced from Twitter and other popular blogs.

3 Challenges observed in CM data

1. **Gathering data**: Data collection is the primary and the most crucial step while dealing with the problem with any neural-network-based approaches (Roh et al., 2021). There is a huge challenge for collecting CM data due to fewer resources and its informal nature of use in limited places. Datasets for low-resource languages like Telugu are challenging to find, making it difficult to build supervised models.

2. **Misspelled words**: Since most of the data for CM comes from informal resources like social media posts, casual blogs, some of them are misspelled (refer col. 3 table 1). This is a significant challenge while building spell agnostic models.

3. **Romanization Variablity**: One other defiance/challenge in CM data apart from spell checking is the variability in romanization output. For example, the Telugu word 'enduku'(Meaning: why) can either be written as 'endhuku' or 'nduku' (For more examples, refer Table 1).

4. **Feature extraction**: Due to misspelled words and romanization variability, popular feature extraction methods like Word2Vec cannot be used due to high variations of the same word. We, therefore, used low-level features (explained in 5.1), which we found to be working well for this task.

5. **Morpheme-level CM**: As explained in section 1 example, Handling morpheme-level CM adds more complexity to the problem as the word is a combination of two words from two different languages.

In the next section, we explain the procedure used to create LID dataset for CM English-Telugu data.

4 Dataset

We created two different types of code-mixed datasets sourced from Twitter[3] and popular blogs[4][5] using Code-mixed language.

As shown in table 1, the Twitter dataset has significant variability in style, whereas the second dataset consists of sentences from articles written by professionals, hence have minimal variability. Table 1 shows the variability in styles of writing the same word in different ways across both the datasets.

For the Twitter dataset, we manually identified 40 user accounts who tweet with CM; these user accounts often tweet on different aspects such as Movies, Politics, and Sports. We used the Twitter API to get 1000 tweets of each user. For the blog dataset, we scrapped chaibisket.com and wirally.com websites which provide high-quality code-mixed content. We manually picked 350 articles containing code-mixed data for this dataset.

The pre-processing of the data involved the following steps:

1. Converting tweets into sentences
2. Removing sentences containing Dravidian characters
3. Removing sentences that contain only English words and only Telugu words
4. Removing sentences with less than five words
5. Removing URLs and other similar tokens

[3]Webpage: https://twitter.com/
[4]Webpage: https://chaibisket.com/
[5]Webpage: https://wirally.com/

Source Word	Blog Dataset	Twitter Dataset
ఎందుకు	enduku	enduku, endhuku, nduku
చెప్పు	chepu	cheppu, chpu, chepu
మీకు	meeku	meeku, meku, miku
hyderabad	hyderabad	hyderabad, hyd, hydbad
correct	correct	correct, crct

Table 1: Table explaining the romanization and spelling variability in both the datasets

Label	Blog	Twitter
English	50891	70751
Telugu	76003	68762
Named Entities	6146	36226
Universal	2608	36281
Total words	135648	212020
Avg Sentence length	14	8.6
Total Sentences	9657	24404

Table 2: Statistics on words in Datasets

Label	Blog	Twitter
English	0.95	0.96
Telugu	0.93	0.94
Named Entities	0.94	0.97
Universal	0.98	0.97

Table 3: Cohen-Kappa Score for Inter-Annotator Agreement.

6. Tokenizing emojis and hashtags

After preprocessing the data, it is then manually annotated into four classes, i.e., Telugu (TE), English (EN), Named Entities (NE), and Universal (Univ). Named Entities include names of people, organizations, and locations. Universal includes punctuation marks, acronyms, emojis, hashtags, and numbers.

4.1 Inter Annotator Agreement

Three people proficient in Telugu dialects and English language and having a linguistic background have annotated the dataset. We calculated Inter Annotator Agreement score using Cohen's kappa score (Cohen, 1960) in order to assess the quality of both datasets. Table 3 reports the Inter Annotator Agreement scores.

5 Methodology

In this work, we consider LID in CM data as a classification problem where we label each word to its corresponding language class. With the advent of deep learning, many popular tasks like Named-entity recognition, Parts-of-speech identification have shown promising results (Singh et al., 2018), (Meftah and Semmar, 2018). In section 5.2 we explain an extensive set of deep learning models for LID in CM data. We show the quantitative results of LID in section 6. We provide the following architectures based on the type of input the model is supplied with to solve this problem:

1. **Word Level Classification (WLC)**(fig. 1): Given a word, we classify it into one of the four classes. This approach does not take advantage of the contextual information of the given word in the sentence. The example given below illustrates the input and output from this approach.
   Input: bagundi (Translation: good)
   Output: TE

2. **Sentence Level word-by-word Classification (SLC)**(fig. 2): Given a sentence, we predict each word's label in the sentence. This approach utilizes the contextual information of the word in the sentence and predicts the output of each word. Below example illustrates the input and output from this approach.
   Input: Sai class ki velli book theesadu . (Translation: Sai went to class and opened a book)
   Output: Sai/NE class/EN ki/TE velli/TE book/EN theesadu/TE ./UNIV

In the following subsections, we explain our approach as a two-fold process: (1) Feature representation (2) Model Training.

5.1 Feature representation

Feature representation plays a crucial role in the training of a deep learning model and increasing its efficacy. The following subsection introduces two types of feature representa-

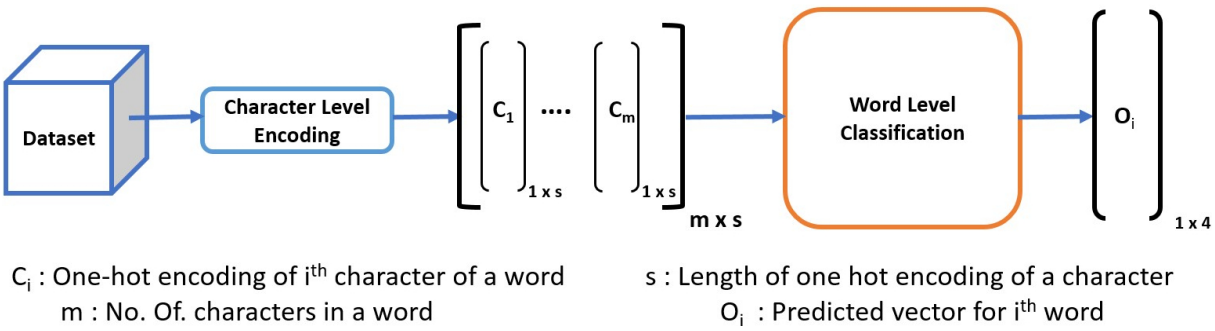

Figure 1: Pipeline for Word Level Classification with Character Level Encoding.

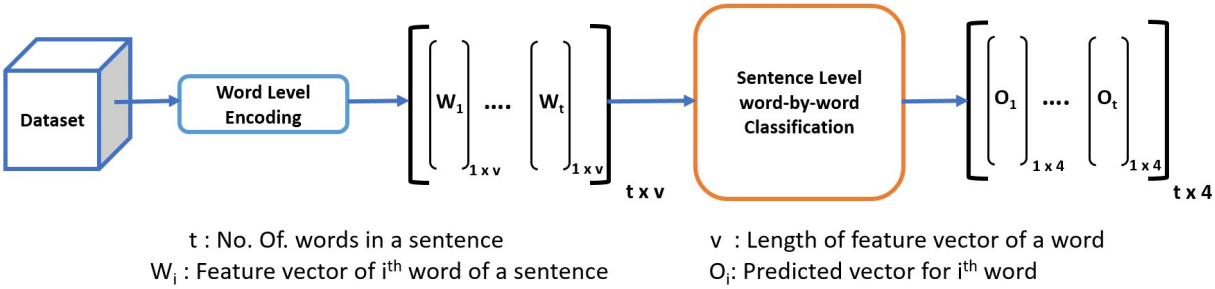

Figure 2: Pipeline for Sentence Level word-by-word Classification with Word Level Encoding

tions that we use in our proposed architectures (WLC, SLC).

5.1.1 Character Level Encoding (CLE)

In this approach, each character of the word is made into a one-hot encoding vector.

$$FCLE : [c1_{1\times m}, c2_{1\times m}, ...cn_{1\times m}]_{n\times m} \quad (1)$$

Where $c1, c2, ...cn$ are one-hot encodings of the characters of a word of length n.

5.1.2 Word Level Encoding (WLE)

In this approach, each word in a sentence is encoded as a vector having the following features:

- **Character N-Grams:** Character N-Grams of the word.
- **TF-IDF:** Term-Frequency and Inverse-Document Frequency of the N-Grams feature vector.
- **Hand-picked features:** The below high level features are chosen to capture semantics of various types of words such as Acronyms, numbers and punctuation.
  - Count of special characters
  - All capital letters
  - Starts with capital letter
  - Number of digits
  - Length of the word

5.2 Deep Learning Models

Deep learning models have been successful in understanding the semantic representations and learning complex tasks in Natural Language Processing (NLP). This section puts forward the various deep learning models we have implemented to solve LID in English-Telugu CM data. To provide an extensive benchmark, we also compare the above models to classical models such as Naive Bayes, SVM, Logistic Regression, and CRF in Table 6.

We divide our deep learning models into two categories. The first set of models namely: LSTM, CNN, RNN, MLP Network are used in our first architecture - WLC (fig 1) and the second set of models namely: BiLSTM + CRF, BiLSTM + LSTM are used in our second architecture - SLC (fig 2).

5.2.1 Word Level Classification (WLC)

In this method, we take each word and extract the features with CLE. The sequence of vectors from CLE is given to a Word Level Classifier, which then classifies the given word into four classes, viz. EN, TE, NE and UNIV. The following models use the pipeline illustrated in fig 1.

- **CNN:** CNNs are mostly used for Images,

but we have seen that CNNs perform on text as well in few applications. For the features, One-Hot encodings of each word are concatenated to get a 2-Dimensional grid, on which the CNNs are applied. We used two Convolutional Layers, one with 32 filters and the other 16 filters.

- **RNN, LSTM:** The major problems faced by CNNs are handling sequential data, considering only current input, and lack of memorization of previous inputs. All these shortcomings are better handled by RNNs. A Recurrent Neural Network works on the principle of saving the output of a particular layer and feeding this back to the input in order to predict the output of the layer. Long Short-Term Memory (LSTMs), on the other hand, is a type of RNNs which prevent the vanishing gradient problem often found in RNNs.

The above-explained models (CNNs, RNNs, LSTMs) utilize positional information of characters either through a kernel convolution(CNN) or through hidden layer propagation(RNNs, LSTM). Thus, we also tried with an MLP model which lacks the above positional knowledge using the feature representation technique explained in 5.1.2.

MLP: A Multi Layer Perceptron (MLP) is a supervised feed forward neural network, which consists of an input, an output layer and few hidden layers. Each layer consists of a set of neurons that receives input from the previous layer and sends output to neurons in the next layer based on activation function. We added three layers of Dense Networks with 256, 64, and 32 neurons for each layer, respectively. Figure 3 shows the pipeline for LID using MLP.

5.2.2 Sentence Level word-by-word Classification (SLC)

In this method, we input an entire sentence and extract features with WLE and classify the output of each word with deep learning models, namely BiLSTM + CRF and BiLSTM + LSTM.

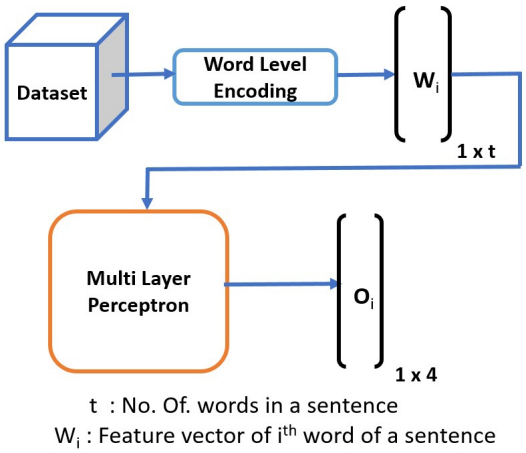

Figure 3: Pipeline for MLP model with Word Level Encoding (WLE) for feature extraction.

BiLSTM + CRF: Bi-directional LSTM was proposed by (Schuster and Paliwal, 1997), is a variant of LSTM which allows data to flow forward as well as backward in time. BiLSTM + CRF improves the performance by giving more context of previous and next occurring data to the model. BiLSTM + CRF have been known to work well in sequence labeling tasks (Poostchi et al., 2018), and hence we have used this model to carry out experimentation on our proposed datasets. Figure 4 illustrates the pipeline for this model.

BiLSTM + LSTM: Each word in a sentence is passed to WLE for feature extraction and then passed to the BiLSTM layer. We then use the hidden outputs of the BiLSTM layer as inputs to the LSTM layer to make the final prediction. Figure 5 illustrates the pipeline for this model.

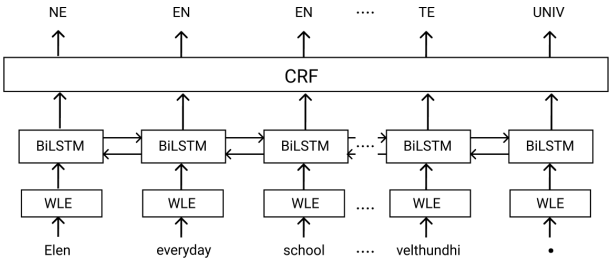

Figure 4: BiLSTM + CRF model.

6 Results

We have put forward an extensive set of deep learning models to tackle LID in English-

Classifier	TE Precision	TE Recall	EN Precision	EN Recall	NE Precision	NE Recall	UNIV Precision	UNIV Recall
MLP	98.84	97.84	97.88	**98.72**	81.42	85.27	94.11	**96.07**
CNN	98.06	97.71	97.21	97.80	83.15	84.11	96.23	92.40
LSTM	98.73	98.37	98.07	98.54	85.62	86.32	95.56	95.13
RNN	92.22	98.27	91.00	95.89	0	0	96.65	58.82
BiLSTM + CRF	98.81	**99.27**	**98.41**	97.95	88.24	85.34	98.13	91.56
BiLSTM + LSTM	**99.04**	99.17	98.21	98.68	**89.98**	**86.40**	**99.37**	92.73

Table 4: Per-Class Precision and Recall metrics on test data of Blog Dataset

Classifier	TE Precision	TE Recall	EN Precision	EN Recall	NE Precision	NE Recall	UNIV Precision	UNIV Recall
MLP	97.69	97.77	97.38	98.37	96.56	94.21	96.39	96.74
CNN	96.76	96.29	96.42	96.61	95.09	94.09	95.34	96.90
LSTM	98.25	97.36	98.03	98.69	97.13	96.63	96.56	97.46
RNN	89.03	95.08	92.34	94.61	96.14	84.52	97.39	97.70
BiLSTM + CRF	**99.52**	**99.35**	**99.14**	**99.17**	**99.21**	**99.53**	**99.35**	**99.00**
BiLSTM + LSTM	98.03	98.80	98.34	98.57	97.55	97.05	99.14	97.76

Table 5: Per-Class Precision and Recall metrics on test data of Twitter Dataset

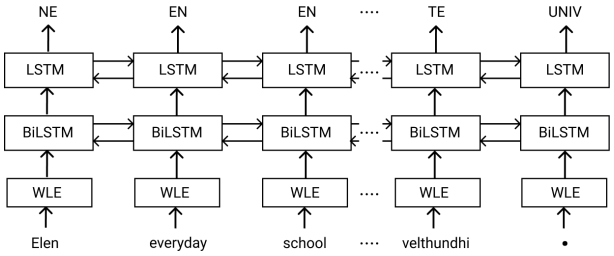

Figure 5: BiLSTM + LSTM model.

Model	Blog	Twitter
Naive Bayes	88.26	86.43
Logistic Regression	94.18	93.59
SVM	90.67	86.85
CRF	**96.15**	**97.23**
MLP	97.54	97.17
CNN	96.98	96.11
LSTM	**97.78**	**97.70**
RNN	91.80	92.57
BiLSTM + CRF	98.35	**99.32**
BiLSTM + LSTM	**98.53**	98.24

Table 6: Model testing accuracy of classical models and Deep Learning models on Blog and Twitter Datasets

Telugu CM data in our present work. To validate these models, we show quantitative results on both the proposed datasets, namely Twitter and Blogs datasets (explained in section 4). We also compare with the existing classical Machine Learning models (refer Table 6). We show Precision and Recall metrics for each class in table 4, 5. It is observed that RNN had faced the problem of vanishing gradients, thus precision and recall of RNN for NE is zero. The BiLSTMs property of propagating contextual information in both directions helps it to have the edge over WLC models like MLP, CNN, and RNN models. From table 4, 5 we see that the BiLSTM + LSTM, BiLSTM + CRF models outperform the other Deep learning models in this task and achieves an improvement in accuracy over the classical models (Baseline: CRF) of around **2.38**%, **2.09**% on the Blog and Twitter data-set respectively. It can also be noted that, though our primary task is to identify Telugu and English words in CM Data, from (from 4, 5) we observe that the precision and recall of the Named Entity class is on the higher side in

SLC (BiLSTM) as compared to WLC models (MLP, CNN, LSTM, RNN). We also see that the accuracy of RNN falls as it suffers from vanishing gradients, which hampers the learning of long data sequences as observed in the Twitter dataset.

One of the few challenges that were encountered was the Romanization of Telugu words and social media acronyms and abbreviations. As explained in section 3, there is no standard way to transliterate the code mixed data, and thus Romanization variability leads to different spellings of the same word. For example, the romanization variability of a single word can be: "eppudu", "epdu", "epudu", "yepudu" (Translation into English: "when"). Similarly, social media chat conversations/tweets using SMS language "you" can be written as "U", "hello" as "helooo", "What's up" as "wassup" etc. All these examples pose a significant challenge while training the LID models in code mixed data.

7 Conclusion and Future Work

In this paper, we have put forward two Telugu-English CM manually annotated datasets which are an order of magnitude greater than the existing dataset, and proposed two architectures for language classification (Telugu and English) in CM data: (1) Word Level Classification (2) Sentence Level word-by-word Classification. We have conducted thorough experimentation and extensive benchmarking across various Deep Learning models and Classical Machine Learning models. We found out that BiLSTM + LSTM, BiLSTM + CRF performs the best among others. We also plan to make our data corpus consisting of low-resourced Telugu and English languages generated from Twitter, and online blogs open-sourced to encourage further experimentation and research. The LID models developed here can also be used in other NLP tasks like Named Entity Recognition (NER), Parts of speech (POS) tagging, and spell-check. The results of the current work are encouraging, and future work will be focused on using sequence level labeling SOTA models like attention in LID. We are also focused on developing spell-checking models which normalize the misspelled and romanization variability as a part of our future work.

References

Sung-Hyuk Cha Bashir Elhaj Ahmed and Charles C.Tappert. 2004. Language identification from text us-ing n-gram based cumulative frequency addition.

Julien Velcin Ciprian-Octavian Truic˘a and AlexandruBoicea. 2015. Automatic language identificationfor romance languages using stop words and di-acritics. *In 17th International Symposium on Sym-bolic and Numeric Algorithms for Scientific Comput-ing (SYNASC)*, page 243–246.

Judith Hayes JohnHughes Clive Souter, Gavin Churcher and Stephen Johnson. 1994. Natural language identification using corpus-based models. *HERMES-Journal of Language and Communicationin Business*, page 13:183–203.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Amitava Das and Bjorn Gamb. 2014. Identifying ¨ languages at the word level in code-mixed indian social media text.

Ted Dunning. 1994. Statistical identification of language. pages 940–273.

Sunil Gundapu and Radhika Mamidi. 2018. Word level language identification in English Telugu code mixed data. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Braj B. Kachru. 1978. Toward structuring code-mixing: An indian perspective.

Nayan Jyoti Kalita and Navanath Saharia. 2018. Language identification on code-mix social text. *Proceedings of the International Conference on Computing and Communication Systems Lecture Notes in Networks and Systems*, page 433–440.

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.

Deepthi Mave, Suraj Maharjan, and Thamar Solorio. 2018. Language identification and analysis of code-switched social media text. *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching.*

Sara Meftah and Nasredine Semmar. 2018. A neural network model for part-of-speech tagging of social media texts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Mostafa Al-Emran Noor Al-Qaysi. 2017. Codeswitching usage in social media: A case study from oman. *International Journal of Information Technology and Language Studies, 16:27-46.*.

Sreekavitha Parupalli, Vijjini Anvesh Rao, and Radhika Mamidi. 2018. Towards automation of sense-type identification of verbs in OntoSenseNet. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 61–66, Melbourne, Australia. Association for Computational Linguistics.

Hanieh Poostchi, Ehsan Zare Borzeshi, and Massimo Piccardi. 2018. BiLSTM-CRF for Persian named-entity recognition ArmanPersoNERCorpus: the first entity-annotated Persian dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Shana Poplack. 1980. Sometimes i'll start a sentence in spanish y termino en espaÑol: toward a typology of code-switching 1. *Linguistics*, 18:581–618.

Yuji Roh, Geon Heo, and Steven Euijong Whang. 2021. A survey on data collection for machine learning: A big data - ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1328–1347.

David Sankoff and Shana Poplack. 1981. A formal grammar for code-switching. *Papers in Linguistics - International Journal of Human Communication*, 14:3–46.

Mike Schuster and Kuldip Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45:2673 – 2681.

Gupta S. Motlani-R. Bansal P. Shrivastava M. Mamidi R. Sharma Sharma, A. 2007. Parsing pipeline - hindi-english code-mixed social media text.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for Hindi-English code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35, Melbourne, Australia. Association for Computational Linguistics.

P. V. Veena, M. Anand Kumar, and K. P. Soman. 2018. Character embedding for language identification in hindi-english code-mixed social media text. *Computación y Sistemas*, 22(1).

Jatin Sharma Kalika Bali Yogarshi Vyas, Spandana Gella and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 974–979.

Kelly Ann Hill Zirker. 2007. Intrasentential vs. intersentential code switching in early and late bilinguals.