Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 3425–3433
Marseille, 20-25 June 2022

MHE: Code-Mixed Corpora for Similar Language Identification

Priya Rani, John P. McCrae, Theodorus Fransen
Data Science Institute, National University of Ireland Galway
Galway, Ireland
{prani1, john.mccrae, theodorus.fransen}@nuigalway.ie

Abstract

This paper introduces a new Magahi-Hindi-English (MHE) code-mixed dataset for similar language identification, where Magahi is a less-resourced minority language. This corpus provides language identification at two levels: word and sentence. This dataset is the first Magahi-Hindi-English code-mixed dataset for similar language identification task. Furthermore, we will discuss the complexity of the dataset and provide a few baselines for the language identification task.

Keywords: Magahi, Code-Mixing, Similar languages, Language Identification

1. Introduction

Language identification has been called a "solved task" (McNamee, 2005) and while this may be the case for many languages, for closely-related languages this is still a challenging task (Goswami et al., 2020). In particular, for many closely-related languages there is a large degree of code-mixing in the typical usage of these languages and as such language identification changes from a sentence- or document-level task into a word-level task. In particular, we focus on the case of Magahi-Hindi-English code-mixed texts, where speakers of Magahi frequently mix their language with the most widely-spoken language in the country, Hindi, or with the international language of English. Magahi is not an official language of India and is often considered as a Hindi dialect; however, the massive growth in and increasing access to social media means that the language is developing a new identity online. There is significant overlap in the vocabulary between Magahi and Hindi, which further adds to the challenge of this task: words may simply be of both languages with no distinction made, or words from one of the language may be used with the morphology and grammar of the other. As such this creates a challenging language identification task for which there exist no corpora which can be used to develop systems. For less-resourced minority languages such as Magahi, developing corpora is therefore a challenging task, and the availability of annotators as well as the linguistic challenges of working with closely-related languages in code-mixed scenarios add substantial obstacles to corpus development.

In this work, we present the first code-mixed corpus for Magahi-Hindi-English based on text collected from social media platforms. The text in our corpus is code-mixed text as is common in user-generated content (Zhang et al., 2018). Code-switching occurs when conversants use both languages together to the extent that they change from one language to the other in the course of a single utterance (Wardhaugh and Fuller, 1986). This linguistic phenomenon is very prevalent in informal communication in a multilingual society where speakers switch between two or more languages and this is also a well-known practice among social media users. The corpus we developed shows a substantially higher degree of code-mixing compared to other corpora, as illustrated by its code-mixing index, yet we show that we have developed robust annotation guidelines that have allowed us to develop the corpus with a very high degree of agreement between multiple annotators.

Further, we present experiments on automatic language identification, which has become one of the prerequisites for natural language processing pipelines and we hope can be a baseline for researchers developing methods for language identification on challenging language combinations such as Magahi-Hindi-English.

Our contributions described in this paper are the following:

- MHE - an annotated Magahi-Hindi-English code-mixed corpus for similar language identification. To the best of our knowledge, this is the first Magahi-Hindi-English dataset for similar language identification, which contains language id at both sentence and word level.
- An extensive analysis of the code-mixing in the dataset. It also provides a comparative study of the complexity of the available code-mixed datasets using the Code-Mixing Index (CMI).
- We present baseline scores for similar language identification on both sentence and word level identification tasks.

2. Literature Review

The Language Identification (LI) task has always been one of the necessities of Natural Language Processing (NLP). In some of the NLP task, such as machine translation, it is a prerequisite factor to determine the source and target language (Barman, 2019). LI has been investigated both linguistically and statistically at various granularities from document level, sub-document

level, sentence level to word level (Barman, 2019). However, the majority of language identification tasks involves well-resourced languages such as Hindi, English, Spanish and French (Hughes et al., 2006). On the other hand, a notable amount of work has been explored by several researchers over the past decades for different pairs of less-resourced and minority languages and language varieties, especially in the social media domain (Barman, 2019; Solorio et al., 2014).

As code-mixing is commonplace in social media content, especially by users from multilingual communities, there has been a surge in research on code-mixed LI. In recent years, this research jumped out of its comfort zone of resource-rich languages to less-resourced and minority languages. Several datasets and LI models have been trained using state-of-the-art methods to improve the results of the code-mixed LI task. King and Abney (2013) used a dataset of 30 languages to perform a language identification task using semi-supervised methods. They have explored several classifiers starting with Naive Bayes for word-level classification and sequence labelling with CRF trained with Generalised Expectation Criteria, receiving the accuracy of 0.93, approximately.

Nguyen and Doğruöz (2013) performed language identification experiments on Turkish and Dutch forum data. They carried out experiments with many models such as dictionaries, logistic regression, and CRF. They found out that language models are more robust than dictionaries, and contextual information is helpful for the task. The work conducted by Castro et al. (2016), tries to discriminate between Brazilian and European Portuguese on Twitter. They use an ensemble method with character 6-grams and word uni- and bi-grams to achieve an accuracy of 0.9271.

The organisation of several shared tasks, such as The First and Second Shared Task on Language Identification in Code-Switched data (Solorio et al., 2014; Molina et al., 2016) and The FIRE Shared Task on Mixed Script Information Retrieval (Sequiera et al., 2015), has attempted to motivate researchers to do more advanced research towards word-level language identification tasks for Indian code-mixed languages. Das and Gambäck (2014) used a dictionary and SVM model with various features such as n-gram, minimum edit distance and word context information to identify languages at the word level in a Hindi-English and Bengali-English dataset.

Despite there being a high interest in discriminating between similar languages in the European, Asian, and Arabic context, there are hardly any similar attempts to identify less resourced Indian languages specifically in code-mixed scenario. Kusampudi et al. (2021) created a corpus of low-resourced English-Telugu data from social media such as Twitter and local blogging sites such as Chaibisket.com and Wirally.com. They conducted their experiments on various baseline models using both machine learning and deep learning models with the highest accuracy score of 98.53 on blogging data and 98.24 on Twitter data, using a BiLSTM+LSTM model.

Given the fact that most of the social media content generated is code-mixed and does not only include majority languages, there is a need for enhancing the basic task like language identification for Indian languages; thus the addition of novel datasets and detection models, especially for similar languages and varieties, would move us one step ahead in solving the task for similar language identification in code-mixed scenarios.

3. Corpus Creation and Annotation

The collected and processed dataset is a mixture of Magahi, Hindi and English. In certain areas on the Indian subcontinent both Magahi and Hindi are spoken, making code-mixing very likely; this phenomenon is indeed commonplace. Being closely-related languages, Magahi and Hindi share a lot of complex and hierarchical relations. At the same time, as is the case everywhere in India, English is one of the primary and frequent languages to be code-mixed with other languages by social media users.

3.1. Data Creation

The data source for the current corpus is YouTube. We collected the data from publicly available comments on YouTube as some of the Magahi speakers are highly active on YouTube. For this task, we selected two YouTube channels, namely, 'Magadhi Boys'[1] and 'Magadh Music' [2]. These two channels are very active and the videos uploaded are on various themes such as folklore, mythology, society, politics, environment, entertainment and many more. The comments on the uploaded videos are very useful for building our corpus, especially the Magahi corpus. The users are motivated to comment on Magahi and thus create Magahi-Hindi-English code-mixed text. Both Roman and Devanagari script are used for comments.

We extracted the comments using a YouTube scraper[3]. The comments were later tokenized for labelling the language id at word level and the extracted comments were given for sentence level identification. Detailed statistics of the dataset are provided in Table 1.

Number	MHE
Sentences	16,784
Words	146,256
Unique words	15,348

Table 1: Statistics of our YouTube Magahi-Hindi-English (MHE) code-mixed dataset

[1]https://www.youtube.com/channel/UCvh5PbwK8I3lyRSQSjsqYwQ

[2]https://www.youtube.com/c/MagadhMusic1

[3]https://github.com/philbot9/youtube-comment-scraper

3.2. Data Annotation

The annotation of the dataset was completed at 4 different phases in order to validate the annotation guidelines and to increase the authenticity of the dataset. Four annotators took part in the annotation task. Two of the annotators were trained linguists, and the other two were language students. Magahi was the mother tongue of all four annotators and all were highly fluent in the three languages (Magahi, Hindi, English). The annotators' age varied from 24-38 years. Annotation was carried out using a simple Excel sheet. Each Excel sheet has two sheets, namely the sentence sheet and word sheet, where the sentence sheet has only sentences, and the word sheet has only word token of the sentences given in the sentence sheet. Annotators were asked to choose the tags from the drop-down option given on the sheets.

3.2.1. Annotation Tagset

This section details the description of the annotation tag set. Any token, whether word or sentence belonging to a particular category, was tagged with the given category tag. For example:

Tag	Word	Translation
MAG	'हमहु'	I also
HIN	'आपसे'	From you
ENG	'Like'	---
H&M	'saal'	Year
OTH	'@', '#'	---
NAME	'Sushant', 'नरेश'	Sushant, Naresh
NUM	'1', '5'	---
ABV	'CM'	Chief Minister
UNK	'Jena'	This is not

Figure 1: Examples of the distribution of the tags in the MHE dataset at word level.

Tag	Sentence	Translation
MAG	**'Bdi achha laglo Sr asne video bnayte rhiya'**	It was an awesome video, sir please make these types of videos more
HIN	**"Kya baat Bhaiya?'**	What's the matter brother?
ENG	**"Best commentry'**	—
UNK	**'Maithili Jena chhai'**	This is not Maithili

Table 2: Examples of the distribution of the tags in the MHE dataset at sentence level

3.2.2. Annotation Scheme

The annotation was carried out using a flat annotation model at two different levels: word and sentence level. The two levels are not associated with each other at the point of annotation. Both word and sentence-level annotation are used to study the complexity of the dataset and thus the challenges of similar language identification. The annotation schemes for the task are discussed below.

Sentence level The annotation requires that the participants annotate the comments as a simple sentence.

- A comment is marked for a particular language if the number of words in that particular language is more or equal to half of the total number of words in the comment. For example:

(1) 'pranam **karit hiyo**, **papa ke bhi** pranam kar **hawai....** *like* to **thokbo** or *subscribe* **bhi karbo'**
Translation : Hello to you and father..... I will definitely like and subscribe.

In the above given Example 1 there are total of 16 words in the sentence out of which 9 are in bold that represent Magahi words, the 2 words in italics represent English and the remaining 5 words are of Hindi. Since the count of Magahi words exceeds the count by 4 words the comment is marked as **Magahi**.

- A comment is marked for a particular language if the sentence gives more weights or has a strong emphasis on the morphological features/inflection of the comment of the particular language. For example:

(2) 'Bhut din bad **aailhu bhiya**'
Translation: Brother you came after so many days

In Example 2, the main verb is (aa**-il-hu**), where the root *aa* is standard in both Hindi and Magahi. However, the inflection markers **-il** and **-hu** are Magahi inflections; thus, the verb is considered a Magahi verb. Since the main constituent of the sentence is marked as Magahi, the whole sentence is tagged as Magahi.

Word level

- Annotation was solely based on the words themselves. For example:

(3) Baal [lang -"HIN"] n [lang -"MAG"] marde [lang -"MAG"] kesh [lang -"MAG"].

As we can observe in Example 3, every word is being marked with the tag [lang -"MAG"] but **"Baal"** is marked [lang -"HIN"] irrespective of the language of the whole sentence.

- A word with Magahi affixes should be marked as Magahi even if the root of the word belongs to or is shared with another language. For example:

(4) 'bahut-**e**' [word lang -"MAG"] [root lang - "H&M"] [affix lang - "MAG"]

Example 4 shows that a word which has Magahi affixes should be marked as Magahi. Belonging to the same language family and being closely related, Hindi and Magahi share many roots. The affixes, however, are specific to either language and thus play a significant role in differentiating the words of the two languages (Rani et al., 2018).

- Loan words - The words which are part of everyday usage by both Hindi and Magahi speakers should be marked according to the context of the given token. To understand the context, annotators should refer to the preceding and/or the following tokens. For example:

(5) khub [lang -"MAG"] **sundar [lang -"MAG"]** lagit [lang -"MAG"] haw [lang -"MAG"]
Translation: You are looking very beautiful.

(6) bahut [lang -"HIN"] **sundar [lang -"HIN"]** lag [lang -"HIN"] rahi [lang -"HIN"] ho [lang - "HIN"]
Translation: You are looking very beautiful.

In Example 5 and 6 we discuss the loan word **"sundar" (beautiful)**. According to the annotation rule, the loan word is marked as Magahi in Example 5 since the preceding and the following words of the token are in Magahi. Similarly, in Example 6 it is tagged as Hindi as the words preceding and following the token are in Hindi.

- If the word is used in both Magahi and Hindi and the context of the word is not clear to determine the exact tag for the token, it should be tagged as **"H&M"**. For example, the word **bana**. In Example 7 the preceding and the following words provide a clear context about the language of the words "bana". As the token "bana" is followed by a Magahi auxiliary **"hu"**, so the annotators tagged the main verb *"bana"* as Magahi. However, in Example 8 one cannot determine the language of the token with the context of the verb,**"Bana"** thus being marked as both, i.e. "H&M".

(7) eka (one) [lang -"MAG"] go (numeral classifier) [lang -"MAG"] film (film) [lang - "H&M"] **bana (make) [lang -"MAG"]** hu (auxilary) [lang -"MAG"] ne (marker) [lang -"MAG"] magahi (Magahi) ["NAME"] **me (in) [lang -"MAG"]**
Translation: Please make one film.

(8) Magahi (Magahi) ["NAME"] **bana (make) [lang -"H&M"]** bahut (more) [lang - "HIN"] **me (in) [lang -"H&M"]**

Social media vocabulary/expressions - Any words or phrases that are part of social media vocabulary/expressions should be marked based on the following criteria.

- Construction of the sentence: If a sentence or a phrase which is very typical of social media vocabulary or expressions and is written in both Devanagari and Roman script or either of the scripts then the annotation will be done based on the particular language frame in which the sentence has been constructed. For example:

(9) like karahoo [lang- "MAG"]
Translation: Please like it

(10) Like and subscribe [lang- "ENG"]

3.3. Annotation Analysis

3.3.1. Inter Annotator Agreement

We calculated the inter-annotator agreement using Krippendorff's α with the help of Krippendorff 0.32 based on the Thomas Grill implementation[4].The agreement was calculated on a subset of 2000 comments. In each phase all four annotators were given a subset of 2000 comments in the sentence-sheet and approximately 19,420 word tokens in the word-sheet. The annotation phase has been categorised into two categories:

1. **Simple data annotation -** annotation phase 1 and 2 were put in this category. The data given for the first category annotation phases were simple linear data were one could infer the context of the token as the each comments were tokenized at word level and arranged vertically. There was no shuffling of the token from one sentence to another.

2. **Reshuffled data annotation -** Annotation phases 3 and 4 were placed in this category. The data for this phase of annotation has been reshuffled after tokenization. One could not infer the context of the token as each token was shuffled against each other and lost its original place in the dataset.

The score of the inter-annotator agreement for each phase of annotation is recorded in Table 3.

[4] https://pypi.org/project/krippendorff/

Phase	Word-level	Sentence-level
Phase-1	0.87	0.89
Phase-2	0.91	0.93
Phase-3	0.83	—
Phase-4	0.89	—

Table 3: Krippendorff's α for inter-annotator agreement of the dataset

Simple data annotation We got an inter-annotator agreement of 0.87 for word-level and 0.89 for sentence-level annotation after the first annotation phase. Even though the agreement score is quite good, there were a few issues in annotating some ambiguous comments and word-tokens. One of the biggest challenges in annotating these cases were the annotation guidelines; thus, we made specific changes in the guidelines. The changes in the tag-set and guidelines are described below:

- Introducing a new tag **"H&M"** described in section 3.2.1.
- Introducing a new annotating rule specifically for social media vocabulary or expression described in the last paragraph of section 3.2.2.

The new annotation tags and guidelines improved the agreement score by 0.04 at word level and 0.11 at the sentence level. Thus, the final inter-annotator agreement scores at the word and sentence level are 0.91 and 0.93, respectively.

Reshuffled data annotation The third phase of annotation with reshuffled data got us the agreement score of 0.83. As the score was lower compared to the previous annotation phase, a new rule was added in word level annotation as the last annotation rule of word level annotation described in section 3.2.2 with Example 7. This new addition improved the inter-annotator agreement by 0.06, getting us the score of 0.89.

4. Code-Switching Analysis

In this section we will provide some descriptive statistics about the dataset which will help us to understand the relatedness between languages and the nature of the dataset.

4.1. Lexical Overlap

As we analyse the distribution of the tags in the dataset shown in the Table 4, we can tell that there is a balanced distribution between Magahi and Hindi, which is 30.78 and 30.19%, respectively. However, the percentage of English tokens in the dataset is relatively low, that is, 19.09%. This distribution of the language tags in the dataset shows that the dataset has many code-mixing instances.

Tag	Count	Percentage (%)
MAG	44,169	30.19
HIN	45,025	30.78
ENG	27,934	19.09
H&M	11,670	7.97
OTH	1610	1.10
NAME	14918	10.19
NUM	731	0.49
ABV	174	0.11
UNK	25	0.01

Table 4: Statistics of the distribution of the tags in the MHE dataset

As we are trying to build dataset for similar language identification, it is necessary to understand the similarity between the languages. Both Hindi and Magahi belong to the Indo-Aryan language family and share the geographic distribution; thus, they have been in contact for a long time and share many linguistic features. Therefore, we started studying similarities between the two languages with the study of lexical overlap. Knowing the lexical overlap would help us understand the complexity of the language identification task and the complexity of the code-mixing.

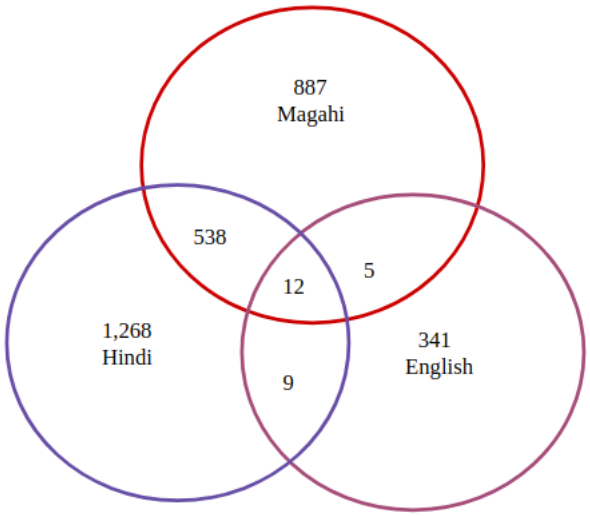

Figure 2: Lexical overlap between the three languages calculated with the unique word tokens over the subset of 8000 comments.

Figure 2 shows the distribution of overlapping tokens. Approximately 30.10% Magahi words overlap with Hindi words, 4.60% overlap with English words and 0.66% of words overlap with both Hindi and English together. It is not surprising that the percentage of lexical overlap between Magahi and Hindi is higher than English and thus we conclude that Magahi has a higher degree of lexical relatedness with Hindi which definitely complicates the language identification task.

However, the involvement of English in the dataset along with the other two languages complicates the challenges of the language identification tasks even more for a code-mixed environment. The complexity of data increases as the number of code-mixed languages increases as there are inconsistencies in labeling certain tokens that are shared among all the three languages. The reason behind these inconsistencies basically depends on two factors:

- Same words across languages - Some words are common across all the three languages for example 'sir', 'like', 'subscribe', 'level', 'comment', 'video' and many more. These are common in YouTube comments and are sometimes marked as English, Hindi or Magahi.
- Phonetic similarity in the spelling - Most of the social media contents are typed using phonetic typing style, which creates same letter representation across the languages for example the letter 'u' represents *that* in Magahi and *You* in English. Similarly, there are many tokens that are responsible for creating inconsistent tags and as such confuse the models in identifying the correct tags for them.

We tried to reduce such errors manually, however these errors are the by-product of code-mixing which include romanisation of the contents and would prevail in the data no matter how much we try to reduce. The only thing that we could do it to make our model more robust to perform well on such raw data.

4.2. Code-Mixing Index

The medium of communication, context of language use, topic, authors, and the languages involved in mixing are some of the factors that determine the characteristics of code-mixing in the datasets (Mave et al., 2018). In order to measure the complexity of the code-mixing in the dataset we calculate the Code-Mixing Index (CMI) (Gambäck and Das, 2014). The Code-Mixing Index is calculated at the utterance level, by finding the most frequent language in the utterance and then counting the frequency of the words belonging to all other languages present in the dataset as illustrated in equation 1.

$$CMI = \frac{\sum_{i=1}^{N}(w_i) - max\{w_i\}}{n - u} \quad (1)$$

Where $\sum_{i=1}^{N}$ is the sum over N languages in the utterance, $max\{w_i\}$ the highest number of words present from any language, *N* the number of languages in the utterance, *n* the number of tokens, and *u* the number of language-independent tokens. We utilize the index to evaluate the level of code-mixing in our MHE dataset over all utterances. The index was calculated using the count of H&M tokens in $max\{w_i\}$, shown in Table 4. The CMI score between the language pairs and MHE is quite large. It is evident from the CMI score of the MHE dataset that the language-mixing rate is relatively high and that we thus have a higher number of code-mixing points. Therefore, we can say that the data is more complex as compared to other datasets and, consequently, acquiring a good performance level for language identification models is much harder.

Language Pair	CMI
English-Bengali (Gambäck and Das, 2014)	24.48
Dutch-Turkish (Nguyen and Doğruöz, 2013)	22.65
Modern Arabic-Egyptian Arabic (Molina et al., 2016)	3.89
Spanish-English (Mave et al., 2018)	22.11
Hindi-English (Mave et al., 2018)	22.22
Nepali-English (Solorio et al., 2014)	20.32
Magahi-Hindi-English	51.54

Table 5: Code-Mixing Index for the different language pair datasets

5. Language Identification Baseline Experiments

In this section we will briefly describe the various models used for the baseline experiments along with the feature set. The data was tokenised at the word level using space as the separators. We do not pre-process the data before or after dividing the data into the training set, validation set and test set with the distribution of 70%, 20% and 10%, respectively.

5.1. Models

SVM: The Support Vector Machine is a machine learning algorithm that maximizes a particular mathematical function with respect to the given dataset (Noble, 2006).The objective of linear SVM optimization problems is to maximize the given equation:

$$max_\alpha \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (x_i x_j) \quad (2)$$

where α_i is the weight of the examples, *x* the input, and *y* the label. After pre-processing the data, we experimented with the most basic input features i.e. character n-grams and TF-IDF, which were created with the help of CountVectorizer[5] and TfidfVectorizer[6] of the Scikit Learn package.

Word-Character LSTM: This model is a replica of the model proposed by Samih et al. (2016) and has also been used by Mave et al. (2018) for a Hindi-English

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

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

and Spanish-English language identification task. The input layer has word and character embeddings. The model has two LSTMs mapped with both character and word representations to the hidden sequence and then these are passed through softmax to compute the probability distribution over all the labels.

Character embedding representations help capture both languages' morphological features and thus reduce the out-of-vocabulary problem. All the three languages, Magahi, Hindi, and English, are morphologically very different. This representation would differentiate the morphological features of the two similar languages, Magahi and Hindi. **Word pre-trained embeddings** were learned on fastText using the monolingual data released during the *The Fifth Workshop on NLP for Similar Languages, Varieties and Dialects* (Zampieri et al., 2018). The word embeddings help us to capture the context of the words that are different for every language. The embeddings were trained on 100 dimensions with a learning rate of 0.025 on 10 epochs and a minimum word-count threshold of 3.

Convolution Neural Network (CNN): the model is described by Zhang et al. (2015), which has a one-character embedding layer and four convolution (CONV1D) layers. One max-pooling layer has been added after each convolution layer for the first three convolution layers. The final convolution layer is followed by one hidden layer, which in turn is followed by one softmax layer. The model accepts sentences and words as the sequence for sentence and word level identification, respectively, and characters as input. The character embedding is a one-hot embedding (1-to-n embedding) where the number of unique characters is *n*. The shape of the filter is one-dimensional in size *k*. The filter slides over the input sequence matrix to create the feature map of dimension $b \times f \times s$ where *b* is the batch size, *f* the number of filters used, and *s* is determined by the formula $m - k + 1$ where *m* is the input size. Stride 1 is used to calculate features based on each character, including spaces and special characters.

UDLDI model: Furthermore we have used the supervised set-up of the UDLDI model (Goswami et al., 2020) both sentence and word level language identification.

Sentence embedding was trained using the inbuilt sentence embedding model which has two parts n-gram character level CNN and the self attention mechanism. The combination of character n-gram and attention weights are multiplied with the CNN feature vector to get the sentence embedding, where 1-dimensional CNN (Zhang et al., 2015) accepts the input sequence as a character sequence as if a sentence has m characters then the input sequence would be $S = [w_1, w_2, ..., w_m]$, where w_i is a character in the sentence (including spaces). The model is illustrated in Figure 3.

Word embeddings were trained by making a slight change in the original model built for the sentence embedding. In this experiment, we converted the inbuilt sentence embedding model to a word embedding model where the combination of character n-grams and attention weights are multiplied with the CNN feature vectors to get word embeddings, 1-dimensional CNN (Zhang et al., 2015) accepts the input sequence as a character sequence as if the sentence has n characters then the input sequence would be $W = [c_1, c_2, ..., c_n]$, where c_i is a character in the words.

The output of the sentence and word embedding model is passed to the fully connected layer and then to the softmax layer to return the probability distribution of all the classes for sentence and word level language identification, respectively.

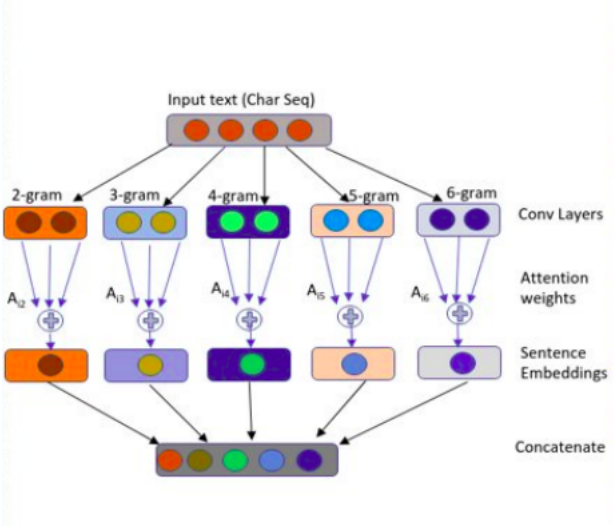

Figure 3: Sentence Embedding model for UDLDI model (Goswami et al., 2020)

CLD2[7] is a very popular and widely used open source language identification tool which supports over 83 languages. The tool is trained on a Naive Bayes classifier.

5.2. Results and Discussion

We use a simple lexicon-based language detection model as the baseline for our language identification system. It uses the most frequent word dictionary method to get a general view of the task. As Magahi does not have a digital dictionary available that could capture the token for our code-mixed dataset, we created the dictionary with an available monolingual dataset (Kumar et al., 2016). After creating the dictionary, we mapped each token in our dataset with the Magahi dictionary and found that only 3.80% of the words were marked as Magahi; however, the data contain 30.19% of Magahi tokens. The rest of the tokens in Devanagari script were marked as Hindi and the re-

[7] https://github.com/CLD2Owners/cld2

maining were tagged unknown. To reduce the errors, we added a new set of code-mixed data to the dictionary, which by default added romanised forms of the lexicon and social media slang to the dictionary, thus increasing the percentage of tokens mapped to 12.60%. Overall, we see varying performance across the models, with some performing much better out-of-sample than others. We computed the F1 score to evaluate the performance of the models on our dataset to account for the imbalance in label distribution (see Table 6). The F1 score of the models are summarised in Table 5.

Models	F1 Score (Word)	F1 Score (Sentence)
SVM (n-grams)	0.54	0.49
SVM (n-grams+TFIDF)	0.43	0.45
LSTM (Mave et al., 2018)	0.74	0.66
CNN (Zhang et al., 2015)	0.77	0.72
UDLDI (Goswami et al., 2020)	0.89	0.84
CLD2[8]	—-	0.68

Table 6: F1 Score of the models at word and sentence level language identification.

The F1 scores using the SVM model for n-gram and the combined n-gram and TFIDF for word and sentence level are 0.54, 0.49, 0.43 and 0.45, respectively. It was found that the SVM model performs poorer than the other models in all cases. At the same time, the F1 scores of the UDLDI model at word and sentence level are 0.89 and 0.84, respectively. Looking at the F1 score of the models, we can observe that the UDLDI model is quite good with "real" social media data as contained in our dataset. When we say "real" data, we mean natural, raw data, not subjected to any pre-processing, having a high level of code-mixing. The CNN model performs slightly better than the LSTM model with F1 scores of 0.77 and 0.72 at word and sentence levels. LSTM performance is acceptable with F1 scores of 0.74 and 0.66, while CLD2 is better than the LSTM model at sentence level identification with an F1 score of 0.68.
The reason behind the poor performance of the models is that these are not adequate to handle the complexity of a high level of code-mixing in the dataset. A few instances of incorrect labelling are discussed below with examples in Figure 4. The examples given in Figure 4 were mislabelled by one or more model. However, the UDLDI model mislabelled only Example 2 and 4, and in most of the cases where the token show phonetic similarity in spellings like Example 2 UDLDI model mislabelled the token. Example 3 was mislabelled by

No.	Word	Gold Label	Label	Translation
1	'go'	MAG	ENG	Numeral Classifier
2	'E'	MAG	ENG	This
3	'बिट्टूआ'	NAME	MAG	Bitu
4	'सितमबर'	NAME	HIN	September
5	'fat'	HIN	ENG	Torn

Figure 4: Examples of errors by the neural models

the other models except UDLDI. As the word in Example 3 consists of Magahi affix the model labeled it as 'MAG' instead of 'NAME'.It is hard to explain why Example 4 was mislabelled by the models. In order to find the reason, we analysed the dataset. Furthermore, we could not find any instance of the name of the months written in Devanagari script; therefore, we concluded that since the words are not in the dataset, the model cannot learn it and thus marked it as 'HIN' rather than 'NAME'. Mislabelling of Examples 1 and 5 could be due to the phonetic similarity of the token in the two languages, which perhaps have confused the model.

6. Conclusion and Future Work

In this paper, we put forward the first Magahi-Hindi-English (MHE) code-mixed annotated dataset for similar language identification. The dataset is annotated both at the sentence and word level with extensive annotation guidelines. We have also conducted thorough experimentation to provide baselines for the dataset with various machine learning and deep learning models.
Our future work will be to come up with a new model which could improve the efficiency of similar language identification for code-mixed scenarios. While the UDLDI model performed quite well, it would need to be improved. This would be possible by studying the nature of code-mixing points in the dataset. Furthermore, we would like to extend our research to enhance the Language Identification task at the morph level with the use of morphological features.

7. Acknowledgements

This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/18/CRT/6223 (CRT-Centre for Research Training in Artificial Intelligence) co-funded by the Irish Research Council under grant number IRCLA/2017/129 (CARDAMOM-Comparative Deep Models of Language for Minority and Historical Languages).

8. Bibliographical References

Barman, U. (2019). *Automatic processing of code-mixed social media content*. Ph.D. thesis, Dublin City University.

Castro, D., Souza, E., and De Oliveira, A. L. (2016). Discriminating between Brazilian and European Portuguese national varieties on Twitter texts. In *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 265–270. IEEE.

Das, A. and Gambäck, B. (2014). Identifying languages at the word level in code-mixed Indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387, Goa, India, December. NLP Association of India.

Gambäck, B. and Das, A. (2014). On measuring the complexity of code-mixing. In *Proceedings of the 11th International Conference on Natural Language Processing, Goa, India*, pages 1–7.

Goswami, K., Sarkar, R., Chakravarthi, B. R., Fransen, T., and McCrae, J. P. (2020). Unsupervised deep language and dialect identification for short texts. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1606–1617.

Hughes, B., Baldwin, T., Bird, S., Nicholson, J., and MacKinlay, A. (2006). Reconsidering language identification for written language resources. In *Proceedings of the fifth International Conference of Language Resources and Evaluation*. European Language Resources Association.

King, B. and Abney, S. (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.

Kumar, R., Ojha, A. K., Lahiri, B., and Alok, D. (2016). Developing resources and tools for some lesser-known languages of india. *Regional ICON (regICON)*.

Kusampudi, S. S. V., Chaluvadi, A., and Mamidi, R. (2021). Corpus creation and language identification in low-resource code-mixed telugu-english text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 744–752.

Mave, D., Maharjan, S., and Solorio, T. (2018). Language identification and analysis of code-switched social media text. In *Proceedings of the third workshop on computational approaches to linguistic code-switching*, pages 51–61.

McNamee, P. (2005). Language identification: a solved problem suitable for undergraduate instruction. *Journal of Computing Sciences in Colleges*, 20(3):94–101.

Molina, G., AlGhamdi, F., Ghoneim, M., Hawwari, A., Rey-Villamizar, N., Diab, M., and Solorio, T. (2016). Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas, November. Association for Computational Linguistics.

Nguyen, D. and Doğruöz, A. S. (2013). Word level language identification in online multilingual communication. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 857–862.

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567.

Rani, P., Ojha, A. K., and Jha, G. N. (2018). Automatic language identification system for hindi and magahi. In Girish Nath Jha, et al., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, may. European Language Resources Association (ELRA).

Samih, Y., Maharjan, S., Attia, M., Kallmeyer, L., and Solorio, T. (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.

Sequiera, R., Choudhury, M., Gupta, P., Rosso, P., Kumar, S., Banerjee, S., Naskar, S. K., Bandyopadhyay, S., Chittaranjan, G., Das, A., et al. (2015). Overview of FIRE-2015 Shared Task on Mixed Script Information Retrieval. In *FIRE workshops*, volume 1587, pages 19–25.

Solorio, T., Blair, E., Maharjan, S., Bethard, S., Diab, M., Ghoneim, M., Hawwari, A., AlGhamdi, F., Hirschberg, J., Chang, A., and Fung, P. (2014). Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar, October. Association for Computational Linguistics.

Wardhaugh, R. and Fuller, J. (1986). An introduction to sociolinguistics.

Marcos Zampieri, et al., editors. (2018). *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Zhang, Y., Riesa, J., Gillick, D., Bakalov, A., Baldridge, J., and Weiss, D. (2018). A fast, compact, accurate model for language identification of codemixed text. *arXiv preprint arXiv:1810.04142*.