


ARTICLE

SwitchNet: Learning to switch for word-level language identification in code-mixed social media text

Neelakshi Sarma* , Ranbir Sanasam Singh and Diganta Goswami

Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati, 781039, India

*Corresponding author. E-mail: s.neelakshi@iitg.ac.in

(Received 22 June 2020; revised 21 April 2021; accepted 26 April 2021; first published online 3 June 2021)

Abstract

Word-level language identification is an essential prerequisite for extracting useful information from code-mixed social media content. Previous studies in word-level language identification show two important observations. First, the local context is an important indicator of the language of a word when a word is valid in multiple languages. Second, considering the word in isolation from its context leads to more effective language classification when a word is borrowed or embedded into sentences of other languages. In this paper, we propose a framework for language identification that makes use of a dynamic switching mechanism for effective language classification of both words that are borrowed or embedded from other languages as well as words that are valid in multiple languages. For a given input, the proposed switching mechanism makes a dynamic decision to bias its prediction either towards the prediction obtained by the contextual information or that obtained by the word in isolation. In contrast to existing studies that rely upon large amounts of annotated data for robust performance in a multilingual environment, the proposed approach uses minimal annotated resources and no external resources, making it easily extendible to newer languages. Evaluation over a corpus of transliterated Facebook comments shows that the proposed approach outperforms its baseline counterparts: classification based on the contextual information, classification based on the word in isolation, as well as an ensemble of the two classifiers.

Keywords: Language identification; Code-mixing; Social media text; Multilingual

1. Introduction

Word-level language identification of code-mixed social media text is crucial for any downstream text processing applications like sentiment analysis, machine translation, etc. Social media data are inherently noisy in nature owing to the presence of irregular and noisy word forms, abbreviations, and creative spellings. This induces challenges to any form of text processing including language identification. Additionally, in a multilingual environment, factors such as code-mixing and phonetic typing (transliteration) further escalate both the importance as well as the challenges associated with language identification. For example, consider the comment on a celebrity page *What a dumdaar performance! [What an impactful performance!]*. In order to produce accurate results, any text-based application like sentiment analysis will require the information that the word *dumdaar* is a Hindi word that is embedded into an English sentence. Further, consider the phrases *stomach ache* and *koyek din ache* [*Few days left*]. Here the language of the word *ache* varies with the context. In the first case (*stomach ache*), it is an English word, whereas, in the latter case, it is a Bengali word in a phonetically typed Bengali sentence (*koyek din ache*). These examples spell out the ambiguities associated with user-generated content in a multilingual environment and motivate the importance of a good language identification system for subsequent text processing.

The examples also illustrate the requirement of language identification at the finer level of words rather than at coarser levels of documents or sentences.

Although word-level language identification has already been addressed in literature, existing studies in word-level language identification exhibit certain limitations. First, most studies do not take into consideration the diverse nature of the vocabulary in noisy code-mixed environments and the challenges thereof for word-level language identification. In a multilingual environment, different social media phenomena such as lexical borrowing, code-mixing, and phonetic typing lead to shared vocabulary across discourse in different languages. While several studies (Bock (2013); Yip and Matthews (2016)) focus on the sociological and linguistic aspects of these phenomena, from the language identification standpoint, not much effort has been directed towards understanding the diversified nature of vocabulary in multilingual social media text and its impact on language identification. In particular, the examples discussed above highlight two major challenges for language identification in a multilingual environment. First, in a multilingual environment, contextual information plays an important role in identifying the language of words, particularly when words are valid across multiple languages. However, when words are borrowed from a different language (e.g., the word *dumdaar* in *What a dumdaar performance?*), taking into consideration contextual information can be misleading. Therefore, the second challenge in word-level language identification is to identify words that are borrowed and deprioritise the contextual information while identifying the languages of such words. Although each of these problems has been addressed separately in literature (Nguyen and Doğruöz (2013); Patro *et al.* (2017)), a single unified framework that resolves both challenges in unison is lacking. In particular, the non-complementary nature of the problems stands as a major obstruction in achieving a single model that addresses both challenges. Therefore, this paper presents a word-level language identification framework that is robust to such multilingual challenges.

In addition to the limitation discussed above, the second limitation of most existing language identification studies is that they necessitate the use of different resources like dictionaries (Das and Gambäck (2014)), annotated data (Nguyen and Doğruöz (2013)), monolingual corpora (King and Abney (2013)), transliteration resources (Singh *et al.* (2018)), etc. Acquiring these resources is in itself an expensive and tedious task and stands as a major drawback in the applicability of the existing methods to newer languages. Therefore, the objective of this paper is to build a language identification framework that is robust to the requirements of a multilingual environment while using minimal resources such that it can easily be adapted to newer languages with even fewer resources.

In our earlier study (Sarma *et al.* (2018)), language identification considering the words in isolation (*global semantic similarity*) and considering the contextual information (*local contextual similarity*) have been explored. The study shows two important observations. First, considering the words in isolation helps in correctly identifying the language of words that are embedded or borrowed into sentences of other languages. Second, the contextual information is helpful in identifying languages of words when words are valid in multiple languages. However, as mentioned above, in practice, a single unified classification framework is desired that can effectively identify both words that are embedded or borrowed into sentences of other languages as well as words that are valid in multiple languages. In this study, we, therefore, further investigate the output of the two classifiers reported in the study (Sarma *et al.* (2018)). We observe that selecting the correctly classified examples from each of the two classifiers can lead to a significant boost in performance. Motivated by this observation, this study proposes to build a dynamic switching mechanism between the classifier built considering the contextual characteristics of words and the classifier built considering the words in isolation. By designing a dynamic switching mechanism, the proposed method attempts to choose the output of one of the two classifiers based on the input characteristics. An alternate way to integrate the information from multiple classifiers is to build an ensemble classifier. The performance of the proposed framework is also compared to that of a neural ensemble framework, details of which are reported in Section 5.

Frameworks similar to the one proposed in this study have also been explored in other tasks; authors in Rei *et al.* (2016) use an attention layer to shift the attention between character and word embeddings to handle out-of-vocabulary and infrequent words in neural sequence labelling models. A similar model is also used in Miyamoto and Cho (2016) to find an optimal combination of character-level and word-level information for their language modelling task. The difference between these models and the proposed model is that while these models use the attention mechanism to choose between different input representations, the proposed model uses a similar mechanism to choose between classifier outputs for the same input. The advantage of the proposed model is that it can be used with any baseline classifiers. In summary, this paper has the following contributions:

- Proposes a word-level language identification framework for code-mixed social media content that can dynamically switch decisions between different classification outputs based on a given input. Experimental results show that the proposed framework is able to achieve better performance compared to the baseline components as well as the neural ensemble framework.
- Proposes the use of non-text-based features for language identification. Experimental results show that the non-textual features yield performance that is comparable to text-based features.

The rest of the paper is organised as follows. Section 2 discusses the related literature in the area. Section 3 discusses the proposed framework. Datasets used and experimental set-ups are discussed in Section 4. Results and observations are discussed in Section 5. Section 6 discusses the conclusion and future works.

2. Related studies

Automatic language identification is a well-explored problem in literature. Studies have focused on different level of granularities – document-level language identification (Cavnar and Trenkle (1994); Yang and Liang (2010)), sentence-level language identification (Carter *et al.* (2013); Wang *et al.* (2015)) and word-level language identification (Das and Gamback (2014); Rijhwani *et al.* (2017)). While most early language identification studies focus on the document-level language identification of regular text like news articles, historical documents, etc., over the last few years, the attention has shifted towards short noisy text in different social media platforms like discussion forums (Abainia *et al.* (2016)), Facebook (Sarima *et al.* (2019)), Twitter (Zubiaga *et al.* (2016)), etc. While sentence-level language identification has also been addressed for social media text, with the ever-growing popularity of social media platforms, word-level language identification has, in particular, garnered much attention and is an active area of research in the current times. The discussion in this section, therefore, focuses on word-level language identification.

Based on the resources used, word-level language identification approaches used can broadly be categorised into (i) classification using dictionaries, (ii) classification using code-mixed annotated data, (iii) classification using monolingual corpora and (iv) classification using transliteration.

A dictionary-based approach for language identification has been used in Barman *et al.* (2014) where the language of the word is classified based on its frequency of occurrence in multiple language dictionaries. Dictionary look-ups for language identification have also been used in Nguyen and Doğruöz (2013). However, considering the variations of word forms in social media content and unavailability of dictionaries for transliterated text, dictionary look-up is not an efficient approach for language identification in terms of coverage as well as performance.

The use of code-mixed data annotated at the word level with the corresponding language information to train word-level classifiers is one of the most commonly used approaches.

Classification-based approaches like SVM, Naive Bayes, and sequence classification-based approaches like CRF have been used in Barman *et al.* (2014) and Gundapu *et al.* (2018) to train word-level classifiers. CRF-based approaches have also been used in Nguyen and Dođruöz (2013), (Xia (2016), Sikdar and Gambäck (2016) and Chittaranjan *et al.* (2014). A CRF classifier combined with post-processing heuristics has been used in Banerjee *et al.* (2014). In addition to addressing word language identification, authors in Das and Gambäck (2014) also introduce a metric called Code-Mixing Index that shows the level of mixing between different languages in a given text. Authors in Vyas *et al.* (2014) also use a CRF-based classifier for word-level language identification, where they also address other tasks like back transliteration, normalisation, and part-of-speech tagging. Word-level language identification and prediction of code-switching points have also been addressed in Piergallini *et al.* (2016). Sequence classification using RNN has been used in Samih *et al.* (2016). A multichannel convolutional neural network (CNN) combined with a bidirectional long short-term memory (BiLSTM)-CRF module has been used in Mandal and Singh (2018). Sequence to sequence models has also been used in Jurgens *et al.* (2017). Instead of sequence models, authors in Zhang *et al.* (2018) propose a two-stage model wherein in the first stage, a distribution over the languages for a given word is predicted, followed by a decoder in the second stage which along with the language distribution predicted in the first stage also takes into consideration the global constraints over the entire sentence.

A drawback of using word-level annotated code-mixed data for training classification models is that they are expensive to obtain. Considering variations in word forms in social media text owing to factors like code-mixing, phonetic typing and spelling variations, in the absence of large-scale annotated data, it may not be possible to capture word and language variations in a highly multilingual environment. Therefore, in an effort to reduce the dependence on annotated data, classification using monolingual corpora in the respective languages has been proposed. Authors in King and Abney (2013) use a collection of monolingual texts and employ weakly supervised and semi-supervised methods for word-level language identification in multilingual documents. Authors in Rijhwani *et al.* (2017) also use unsupervised models using monolingual corpora and HMM for word-level language identification. Instead of using real code-mixed datasets, authors in Gella *et al.* (2014) also create a synthetic code-mixed language identification dataset from a collection of monolingual text. However, a drawback associated with using monolingual corpus for word language classification is that it is not easy to obtain a clean monolingual corpus for transliterated text. Considering that a large section of Indian social media conversations is in transliterated form, these approaches may not be feasible in many scenarios.

Most of the studies discussed above make use of only word-level information for language identification. However, character-level information has also been found useful. Capturing character-level information is associated with two advantages. First, it can be useful in handling unseen word variations. Second, it is most likely that character sequences of different languages will have different structural properties which can be used to disambiguate languages of words. Character embeddings and subword unit information have been used to capture this information in Jaech *et al.* (2016) and Mave *et al.* (2018). In a slightly different approach, authors in Singh *et al.* (2018) use a transliteration model to transliterate romanised script to Devnagiri script. They use RNNs to train a character language model for each language. Given an annotated corpus, the output of the language models is combined with other features to train a word-level language classifier. In addition to character n-gram information, phonetic information has also been used for word language identification in Das *et al.* (2019).

Apart from individual studies on word-level language identification, several survey articles (Jauhiainen *et al.* (2019) and Garg *et al.* (2014)) dedicated to language identification also exist in literature. A number of shared tasks (Solorio *et al.* (2014) and Molina *et al.* (2016)) have also been organised to address word-level language identification problems. Apart from word-level language identification, other closely related studies include analysing different aspects of code-switched data (Rudra *et al.* (2019)), prediction of code-switching points from multilingual

communications (Papalexakis *et al.* (2014)), predicting foreign language usage in social media posts (Volkova *et al.* (2018)), language informed modelling of code-mixed data (Chandu *et al.* (2018)), predicting the presence of matrix language (Bullock *et al.* (2018)) and predicting likelihood of word borrowing in social media (Patro *et al.* (2017)). Recently subword-level language identification has also been addressed in literature. Although it is beyond the scope of this study, some of the recent studies around intra-word code-switching involve the work in Mager *et al.* (2019) and Nguyen and Cornips (2016).

While most of the existing studies report quite satisfactory performance, they have certain disadvantages associated that have also been pointed out in the above discussion. First, developing word-level annotated data is expensive both in terms of time and effort and subject to human error. Second, most methods that avoid the use of word-level annotated data use other resources like dictionaries, monolingual corpora, transliteration tools, etc. Therefore, these approaches are only applicable to languages that are well equipped with these resources. Also, very often, these resources are available in the native scripts of the languages. The use of transliterated text for posting content on social media makes these resources unusable. This paper, therefore, focuses on presenting a method for word-level language identification that uses minimal resources such that it can quickly be adapted for any language. This also makes the proposed approach particularly suitable for low-resource languages. The proposed approach discussed in Section 3.3 is motivated by the observations in Sarma *et al.* (2018).

3. Proposed methodology

This section describes the proposed model to address word-level language identification over code-mixed social media text in a multilingual environment. The objective is to build a dynamic switching mechanism between two classifiers that consider the word in isolation (described as *global semantic similarity*) and the word with its contextual information (described as *local contextual similarity*) (Sarma *et al.* (2018)). Before discussing the details of the proposed methods, we briefly discuss the nature of the dataset used in this study to help the reader to understand the proposed method with ease. Word-level language identification has considered datasets annotated in two different levels: *annotated at the word level* and *annotated at the sentence level*. Word-level annotation is expensive compared to sentence-level annotation. Therefore, word-level language identification using word-level annotated data may not be suitable for low-resource languages. Like in our earlier study (Sarma *et al.* (2018)), this paper considers the same sentence-level annotated dataset. While using sentence-level annotations, the vocabulary set may be divided into two disjoint subsets *resolved set* \mathcal{R} and *unresolved set* \mathcal{U} . The *resolved set* \mathcal{R} are those words that occur only in sentences of a particular language. Therefore, they can be assumed to belong to the same language as the language of the sentences in which they are occurring. The *unresolved set* \mathcal{U} , on the other hand, are those words that occur in sentences of multiple languages. Therefore, the language of these words cannot be confirmed using the sentence-level language information alone. So, the task of language identification can be considered as the task of identifying the underlying languages of the words in the *unresolved set* \mathcal{U} using the words in the *resolved set* \mathcal{R} . The words in the *resolved set* \mathcal{R} are used as training samples in building classifiers. The classifiers proposed in Sarma *et al.* (2018), that is, considering word in isolation and considering contextual information of words are considered as the baseline classifiers in this study. Details of these classifiers are briefly discussed below.

3.1. Word-level language classification considering word in isolation (*global semantic similarity*)

This approach assumes that taking into consideration information about the target word alone is sufficient to identify the language of a word. This approach is based on the target word only irrespective of the context in which the target word appears. Therefore, all occurrences of a given

word form are identified as belonging to the same language. This method, therefore, works well when the percentage of words valid across multiple languages is very low in the corpus.

In this study, the target word is represented by a word embedding vector that is obtained by training word embedding models (skipgram) over a large corpus of code-mixed text. Word embedding models like word2vec make use of the co-occurrence information of words to generate low-dimensional vector representations of words. The resultant vectors capture the semantic and syntactic similarities of words. As a result, similar words lie close together in the projected vector space. The use of these word representations for representing the input to language classification models is driven by the intuition that the word representations obtained by training word embedding models over a corpus of code-mixed text are capable of capturing the language characteristics of the words. As such, words in the same language lie closer in the projected embedding space than words in different languages. Therefore, the language of a target word can be identified by finding its similarity to other words whose languages have already been identified. With words in the resolved set \mathcal{R} as training samples and their vector representations as feature vectors, a classifier is trained. The trained classifier is then used to identify words in the unresolved set \mathcal{U} .

We experiment with four different popularly used classification frameworks – Convolutional Neural Networks (CNNs), Bidirectional Long Short Term Memory Networks (BiLSTMs), support vector machines (SVMs) and logistic regression (LR). The detailed experimental set-up of these classifiers is discussed in Section 4.1. This approach is called the global semantic similarity based approach because the classification is based on the representation of the words in the global embedding space of the corpus.

3.2. Word-level language classification using contextual information (local contextual similarity)

In a multilingual environment, context (preceding words and succeeding words) often helps in identifying the language of a target word, that is, the language of the context and target words may often be the same. Contextual information is particularly important when the languages under consideration contain shared vocabulary, whether due to inherent similarities between languages or due to other phenomena such as phonetic typing or creative writing. Therefore, in the local context similarity-based classifier, context information is also considered while classifying the language of the target words. The target word is represented by a vector which is the concatenation of the vector representations of the target word and words in the context in order of their occurrences.

Like in global similarity, vector representations are obtained by training word embedding models over a large corpus of code-mixed text. Classification frameworks including CNN, BiLSTM, SVM and LR are built over the concatenated vector representations and trained using words in \mathcal{R} as training samples. The detailed experimental set-up of these classifiers is discussed in Section 4.1. This approach is called the local contextual similarity-based approach because the classification is based on the target word as well as the local context in which the target word is occurring.

Both the above classification approaches are associated with respective advantages and disadvantages. The global semantic similarity-based approach, being based only on the embedding vector of the target word, all occurrences of the same word are assigned the same language label. While this approach works well when words are borrowed or embedded into sentences of other languages, this approach fails when the word is valid in multiple languages. On the other hand, considering the contextual information helps in correctly identifying languages of words that are valid in multiple languages. In a noisy multilingual environment, it is desirable that the model learns to prioritise between the words in isolation and the contextual information depending upon a given input. The proposed model discussed below attempts to achieve this by using a feed-forward neural network combined with the output from the individual classifiers which we refer to as the baseline classifiers with respect to the proposed framework.

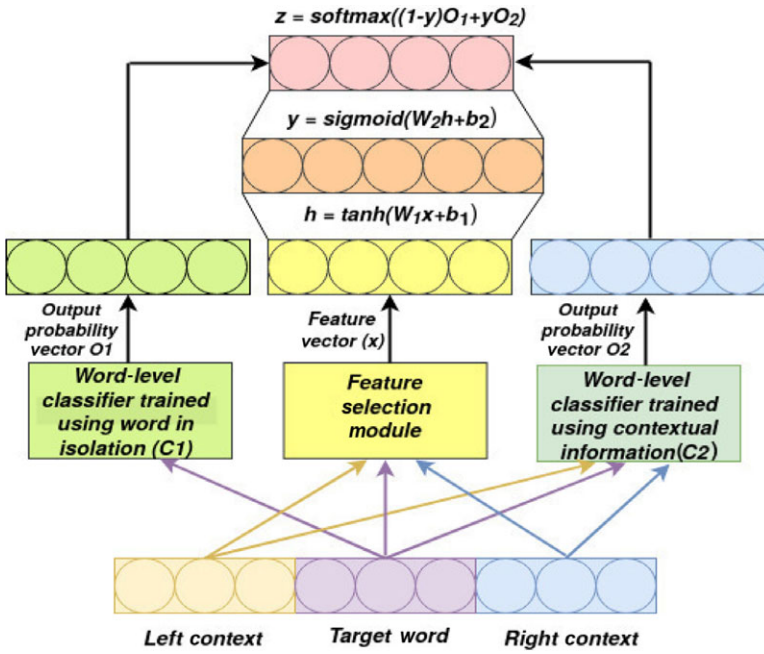


Figure 1. Proposed word-level classifier combining output word-level classifiers built using word in isolation (C1) and using contextual information (C2).

3.3. Proposed classification framework

The proposed approach uses a feed-forward network to dynamically switch decisions between the *global semantic similarity* and *local contextual similarity* for any given input. Figure 1 shows the proposed model architecture. Let C1 represents the *global semantic similarity*-based classifier and C2 represents the *local contextual similarity*-based classifier. The output from C1 and C2 are vectors representing the probability distribution over the languages for the target word. Let these probability distributions be represented by \mathbf{o}_1 and \mathbf{o}_2 , respectively. Let \mathbf{x} be the feature vector representing the target word. This \mathbf{x} could be the textual content of the input. It can also be other non-text-based features representing the target word and its context. The different representations of \mathbf{x} are discussed in detail in Section 3.4. The input vector \mathbf{x} is passed to a fully connected feed-forward neural network with one hidden layer. Let W_1 be the parameter matrix between the input layer and the hidden layer and W_2 be the parameter matrix between the hidden layer and the switch layer. The output vector of the switch layer \mathbf{y} is normalised with a sigmoid operation. The expressions for the forward pass are shown below:

$$\mathbf{h} = \tanh(W_1^T \mathbf{x} + \mathbf{b}_1) \tag{1}$$

$$\mathbf{y} = \sigma(W_2^T \mathbf{h} + \mathbf{b}_2) \tag{2}$$

The vector \mathbf{y} in Equation (2) acts as the switch, the purpose of which is to switch the output of the classifier between \mathbf{o}_1 and \mathbf{o}_2 based on the given input \mathbf{x} . This is shown in Equation (3) where \odot represents element-wise product:

$$\mathbf{z} = \text{softmax}((1 - \mathbf{y}) \odot \mathbf{o}_1 + \mathbf{y} \odot \mathbf{o}_2) \tag{3}$$

If $|\mathbf{y}|$ is high, that is, the values in \mathbf{y} are close to 1, the first component of the sum in Equation (3) reduces to 0 and the final output of the classifier is biased towards \mathbf{o}_2 . Similarly, if $|\mathbf{y}|$ is low, that is, the values in \mathbf{y} are close to 0, the final output of the classifier is biased towards \mathbf{o}_1 . Therefore,

the objective of the proposed framework is to train the network in such a manner that the values in \mathbf{y} range in between 0 and 1 according to whether the final desired output should be biased towards \mathbf{o}_1 or \mathbf{o}_2 . The proposed model consists of three components – the two baseline components and the feed-forward network that acts as the switching mechanism between the baseline components. Instead of training the three components end-to-end, each of the components is trained separately, and the outputs from the pre-trained baseline models are selected based on the output of the proposed switch network, that is, \mathbf{y} . This enables the proposed framework to be used with any baseline component. For learning the parameters of the switch network, the representation vector of the input text \mathbf{x} is considered as the only input to the network. The output vectors \mathbf{o}_1 and \mathbf{o}_2 of the component classifiers are used while estimating the model output \mathbf{z} as defined in Equation (3). Let $\tilde{\mathbf{z}}$ be the ground truth vector (one hot vector) associated with the input \mathbf{x} . Cross-entropy between $\tilde{\mathbf{z}}$ and \mathbf{z} has been used as the loss function for learning the model parameters W_1 and W_2 as defined below:

$$E = - \sum_{i=1}^C \tilde{z}_i \log(z_i) \quad (4)$$

where C is the number of classes, that is, the number of languages. The training procedure for the component classifiers are discussed in Section 4.

3.4. Input representation

The goal of the proposed framework is to dynamically switch between the baseline classification outcomes depending on the given input. This study proposes different features that can be used to represent a given input text to train the proposed framework to make an effective switching decision. These features are discussed below. These features are generated as a part of the feature selection module shown in Figure 1. Figure 2 describes in detail the processing of the input text to generate input vectors of different components. The generated input vector \mathbf{x} is fed as input to the switch network. The following subsections describe the generation of the vector \mathbf{x} which is used as input to the switch network.

3.4.1. Content features

The objective of the content features is to use textual information to identify the language of a word. The textual information considered in this case comprises the target word and the surrounding words. Therefore, the input vector \mathbf{x} , while considering content features, is the same as that of the context-based component classifier, that is, the target word along with its context (preceding and following words) is fed as input to the proposed classifier. The intuition is that the word, along with its context, will be able to capture switching characteristics between the two classifiers. Pre-trained word embeddings obtained by training a word embedding (skipgram) model over a large code-mixed corpus are used to represent the input. While feeding as input to the classifier, the word embeddings of the target word and the words in the context are concatenated in the order of their occurrence (Figure 2(a)).

3.4.2. Neighbourhood based features

The content features make use of the textual information directly for identifying the language of a target word. However, the noise present in the social media data, like irregular and ambiguous word forms, leads to limitations in the performance of the text-based features. Therefore, the goal of the neighbourhood-based features is to extract non-textual information from the data that can overcome the limitations of the text-based features.

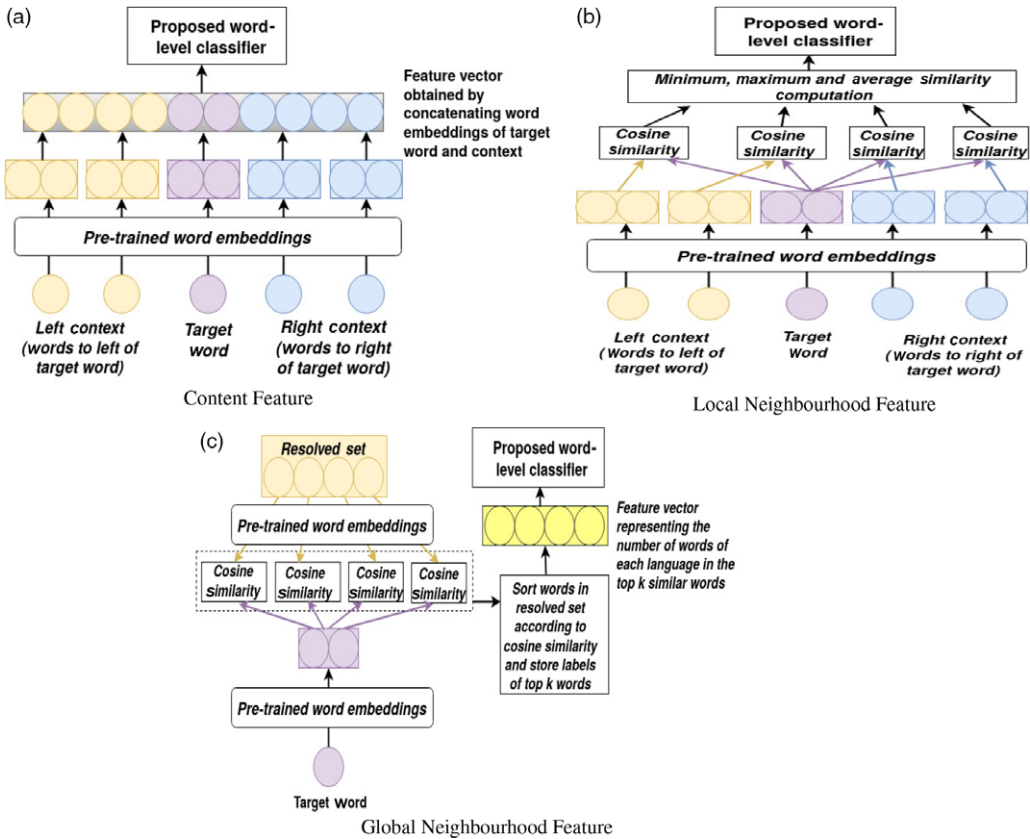


Figure 2. Generation of feature vectors from input text using different components.

The neighbourhood-based features proposed in this study attempts to encode the semantic relation of the words with other words in the corpus into a fixed-sized vector. Therefore, the neighbourhood vector is a fixed-sized vector representing structured information about the input in contrast to the content features that are represented by the word embeddings that accommodate detail semantic information about the word and its context.

This study proposes two neighbourhood-based features as discussed below. The intuition is that the neighbourhood characteristics of a word can be an important indicator of whether the word is a borrowed word or is a word that is valid in multiple languages. These approaches are discussed in detail below:

- Local neighbourhood:** The local neighbourhood-based information tries to encode the semantic information between the target word and its local context (preceding and following words). The intuition is that the semantic distance of the target word with its context can be a good indicator of whether the target word and the words in the context belong to the same language. In other words, if a target word belongs to the same language as its context, its distance from other words in the context will be lesser compared to when the target word belongs to a different language and is embedded or borrowed into the current context. Accordingly, the classification output can be biased towards the prediction obtained considering the contextual information or prediction obtained considering the word in isolation. This approach is therefore called the local neighbourhood-based approach because it considers the local neighbourhood of the target word.

The local neighbourhood is represented by a vector comprising the minimum, maximum and average distance of the target word from the words in the context. The distance is given by the cosine similarity between the embedding vectors of the words. This is shown in Figure 2(b). The intuition is that a word will be semantically more similar to words in the same language than words in other languages. Therefore, when a word is borrowed or embedded into a sentence of another language, its distance from the context would be higher compared to instances when the target word and the context words are in the same languages.

Let \mathbf{t}_i be the embedding vector of the i^{th} word in a sentence. Then, the t_{i-j} and t_{i+j} denote its j^{th} left and right neighbours, respectively. The input vector \mathbf{x} to the switch network is defined by a vector with three elements as follows :

$$x_0 = \min (\{s | s = \text{cosine}(\mathbf{t}_i, \mathbf{t}_{i+j}) \sim \forall_j, -k \leq j \leq k, j \neq i\}) \quad (5)$$

$$x_1 = \max (\{s | s = \text{cosine}(\mathbf{t}_i, \mathbf{t}_{i+j}) \sim \forall_j, -k \leq j \leq k, j \neq i\}) \quad (6)$$

$$x_2 = \frac{1}{2k} \sum_{\substack{j=-k \\ j \neq i}}^k (\text{cosine}(\mathbf{t}_i, \mathbf{t}_{i+j})) \quad (7)$$

where k denotes the number of words in the left and right neighbourhood.

- Global neighbourhood:** The global neighbourhood tries to encode the semantic information of the target word with respect to its neighbours in the global embedding space. The global neighbourhood of the target word is represented by a vector that represents the k -nearest neighbours of the target word in the global embedding space. The intuition is that if a word is valid across multiple languages, the neighbourhood of this word in the global embedding space will consist of words of different languages. On the other hand, if a word is valid only in a single language and its occurrences in sentences of other languages are the result of embedding/borrowing, the global embedding space of such words will comprise all words of the same language. Therefore, while determining the language of a target word, if the neighbourhood of the target word is dominated by words in a single language, the output from the classifier considering the word in isolation should be given preference. On the other hand, if the neighbourhood is composed of words from different languages, then while determining the language of the word, the output from the classifier considering contextual information should be given priority. The global neighbourhood vector \mathbf{x} of a word t is the distribution of the languages in its neighbourhood. Let \mathcal{V}_k be the set of top k most similar words to the target word t and let \mathcal{L} denote the set of languages under consideration. Then the elements of the feature vector \mathbf{x} that denotes the global neighbourhood of the target word t consists of the number of words in each language in the set \mathcal{V}_k . For example if $|\mathcal{L}|=4$, then the feature vector representing the global neighbourhood of the term t is given by $\mathbf{x} = (x_1, x_2, x_3, x_4)$ where $x_i, i = 1 \dots 4$, is the number of words in \mathcal{V}_k that belong to language L_i . The feature vector \mathbf{x} is then used as input to the switch network. As discussed in Section 4, the switch network is built considering only words in the resolved set \mathcal{R} .

3.4.3. Combination of content and neighbourhood-based features

The aim is to represent an input by a combination of its content features and neighbourhood features. The idea is to explore the advantages of combining the detailed semantic information represented by the content features with the fixed-sized structural information represented by the local and global neighbourhood. We explore two ways of combining the features. In the first approach, all three sets of features, that is, the text features, the local neighbourhood features and the global neighbourhood features are merged in the hidden layer. In the second approach, the above sets of features are merged in the hidden layer following a selective feature dropout

Table 1. Description of the dataset annotated at the sentence level

Language	#Sentences	Avg length
Assamese (as)	5198	15 words
Bengali (bn)	1594	12 words
Hindi (hn)	663	12 words
English (en)	21,531	24 words
Total	28,986	–

mechanism proposed in Zhang *et al.* (2018). In the original paper (Zhang *et al.* (2018)), selective feature dropout is implemented by randomly setting the feature group values to zero with 50% probability. However, in the current study, the feature dropout strategy is implemented by alternatively setting a very high dropout (90%) between each feature group and the hidden layer during the training iterations.

4. Dataset and experimental set-up

This study uses the same dataset reported in Sarma *et al.* (2018). This dataset is a collection of code-mixed transliterated Facebook comments collected from a highly multilingual environment using Facebook Graph API. A total of 409,168 user messages have been collected, of which 28,986 messages have been manually annotated with the corresponding language information in the sentence level. Details of this dataset are shown in Table 1. It consists of four languages: Assamese, Bengali, Hindi and English. While identifying Facebook channels for generating the dataset, the following considerations have been taken into account.

- The dataset is collected from a highly multilingual environment where there are chances of users participating in multilingual conversations
- Languages under consideration are such that there is overlapping vocabulary between different languages
- All languages share the same script (Roman script)

Two annotators familiar with all the languages are deployed for annotation. The same set of words are annotated by both annotators. In the end, samples that receive the same annotation by both annotators are retained. The following guidelines have been followed while annotating the messages:

- Messages that contain all words of the same language or that contain words borrowed or embedded from other languages without affecting the underlying language sense are considered monolingual sentences and are assigned the corresponding language label.
- Messages that comprise two or more sentences/fragments in different languages are considered multilingual and are not considered for annotation. For example, the sentence *Movie acchi thi [Movie was good]*, is considered a monolingual sentence (Hindi in this case) and the sentence *Story was good but movie bohot lambi thi [Story was good but movie was very long]* is considered a multilingual sentence.
- Messages that do not belong to either of the four languages are labelled as unknown and are not considered in the current study

Table 2. Description of word-level annotations obtained using sentence-level annotations

Language		Total words
Resolved	Assamese (as)	26,588
	Bengali (bn)	5826
	Hindi (hn)	1679
	English (en)	98,464
Unresolved		495,084

Table 3. Description of the dataset used to train and evaluate the proposed switch network

Language	Training (#Words)	Evaluation (#Words)
Assamese (as)	239	10,132
Bengali (bn)	168	4720
Hindi (hn)	105	6464
English (en)	268	12,859
Total	780	34,169

Using the sentence-level dataset, as discussed in Section 3.3, *Resolved* and *Unresolved* word sets are created. Table 2 shows the number of words in resolved and unresolved sets. The words in the resolved set are used to build the baseline component classifiers.

Further, to train the switch network, a small number of words from the resolved set \mathcal{R} are chosen for manual annotation. The goal for manual annotation is to correct any wrong labels in \mathcal{R} that may have occurred while propagating the language information from sentence level to word level. Furthermore, if both the baseline components result in the same predicted label, then the decision of the switch network is not crucial. The prediction from any one of the baseline components may be chosen as the final prediction. However, if the predictions from both the baseline components are different, then the switch network must learn to make the correct choice between the baseline components based on the input characteristics. Therefore, while choosing training samples for the switch network, only samples where there is a disagreement between the outcomes of the two classifiers are chosen. An evaluation dataset annotated at the word level is created to evaluate the performance of the proposed framework. The evaluation dataset is described in Table 3. It is important to note that considering that the challenge is to disambiguate languages of words in the *unresolved set* \mathcal{U} , the evaluation set only consists of words from \mathcal{U} . The annotation guidelines for creating the word-level annotated dataset are as follows:

- Words are annotated according to the context. The same word in different context may belong to different languages. For example, the word *ache* in the phrases *thik ache* and *stomach ache* are annotated as Bengali and English, respectively.
- Named entities and universal expressions cannot be assigned any language. Instead they are assigned two different class labels *ne* and *amb*, respectively. These words are currently excluded from the evaluation set.

4.1. Experimental set-up

All the data collected go through a pre-processing module where all words are converted to lowercase, and special symbols and numbers are removed. The entire collection of messages (409,168 messages) are used to train a *skipgram* model to generate the word embeddings. The skipgram model is trained for 100 iterations considering window size as 3 and embedding dimensions as 50.

To train the baseline classifiers, four popularly used classification frameworks – CNN, BiLSTM, LR and SVMs are used. For the neural network-based classifiers (CNN and BiLSTM), the pre-trained word embeddings obtained from the skipgram model above are used to represent the input. The CNN classifier is made up of a single convolution layer followed by a max-pooling layer and a fully connected output layer. Fifty filters each of sizes 2, 3 and 4 are used for training the classifiers using contextual information, and 50 filters of size 1 are used for training the classifiers using words in isolation. The BiLSTM classifier also consists of a single BiLSTM layer with 100 units followed by a fully connected output layer. Both the CNN and BiLSTM classifiers are trained using Adam optimiser, and training is monitored through validation loss. Twenty percentage of the training data is set aside as validation data. In the case of the LR and SVM classifiers, the pre-trained embeddings are used as features. For training the classifier using the word in isolation, only the embedding of the target word is fed as input, while for training the classifier using contextual information, the embedding vectors of the words in the context and the embedding vector of the target word are concatenated in order to be used as features. All hyper-parameters are tuned using the validation data described above. The SVM classifier uses a rbf kernel.

The switch network consists of a hidden layer with 100 units and an output layer with the number of units corresponding to the number of languages (i.e., 4). The feature vectors generated using strategies discussed in Section 3.4 are used as input. The text features are constructed considering a window size of 3 on either side of the target word. The local neighbourhood feature is constructed considering a window size of 3 on either side of the target word. The global neighbourhood feature is constructed considering the top 10 neighbours from the resolved set \mathcal{R}^a .

5. Results and observations

This section discusses the results and observations obtained using different experimental set-ups. First, the performance of the baseline classification set-ups is discussed, followed by the performance of the proposed framework using different features.

5.1. Performance obtained using the baseline classifiers

The aim of the proposed framework is to combine the advantages of the classifiers using *word in isolation* and using *contextual information*. With respect to the proposed framework, we refer to these classifiers as baseline classifiers. In this section, we analyse in detail the performance of the baseline classifiers followed by the performance of the proposed framework in the following sections.

5.1.1. Performance obtained considering word in isolation

Table 4 shows the average and the language-wise performance obtained using the baseline classifiers. Considering word in isolation, best macro- and micro-average F-scores of 76.56% and 80.05% are obtained using SVM. The considerably lower F-scores can be attributed to the nature of the classification set-up. In the classification set-up considering words in isolation, words are

^aSince there is not enough word-level annotated data to use as validation data, for the content feature and the local neighbourhood feature, the value of k has been set to 3 as in the baseline set-up, and the value of k for the global neighbourhood has been set to 10 as the performance started converging around $k = 10$

Table 4. Language-wise F-scores and average macro-Fscore (MacF) and micro-Fscore (MicF) obtained by the baseline classifiers

Classifier	Baseline (word in isolation)						Baseline (word with contextual information)					
	as	bn	en	hn	MacF	MicF	as	bn	en	hn	MacF	MicF
CNN	78.72	55.55	89.41	70.40	75.40	78.73	86.27	87.04	84.58	89.52	87.48	86.61
BILSTM	78.33	55.18	89.55	68.67	75.17	78.44	85.63	86.67	84.63	87.48	86.50	85.70
SVM	80.30	57.07	90.90	77.38	76.56	80.05	86.22	86.54	85.21	89.63	87.25	86.50
LR	79.14	55.54	90.96	74.55	75.42	78.91	85.55	85.29	85.11	86.95	86.23	85.58

represented by the word embedding vectors only that yield a single representation for a particular word form. Therefore, all occurrences of the same word are represented by the same feature vector and hence predicted as belonging to the same language. While this is desirable for embedded or borrowed words, for example, the word *movie* in *movie was good* and *movie acchi thi [movie was good]*, it leads to misclassifications in case of words that are valid in multiple languages for example, *tune* in *tune kya kaha [what did you say]* and *tune into my channel*. Therefore, this approach works well when the text consists of large number of embedded or borrowed words but relatively fewer words that are valid across multiple languages.

5.1.2. Performance obtained considering contextual information

From results reported in Table 4, it is seen that considering contextual information, best macro- and micro-average F-scores of 87.48% and 86.61% are obtained using CNN classifier. Classification considering contextual information takes into consideration the fact that words in different contexts can belong to different languages. Therefore, each word is represented by a concatenation of the embedding vectors of the target word and the words in the context. In comparison with classification using words in isolation, it is observed that, on average, across different classifiers, considering the contextual information for classification show better performance. However, upon analysing the classification results further, we obtain the following important observation. It is observed that using classifiers built considering words in isolation, the percentage of embedded or borrowed words correctly classified is 96.17%, while the percentage of legitimate words correctly classified is 76.59%. On the other hand, using the classifier built using contextual information, the percentage of embedded words correctly classified is 86.98%, while the percentage of legitimate words correctly classified is 91.22%. From this analysis, it is clear that although considering contextual information helps in improving the overall classification performance, there is a major drop in performance in terms of language identification of the embedded or borrowed words. This observation has also been reported in Sarma *et al.* (2018) and serves as a major motivation for the current study which attempts at combining the advantages of both the baseline classifiers.

5.1.3. Error analysis of the baseline classifiers

With the motivation stated above, we attempt to estimate the best performance that can be achieved if a correct choice is made between the outputs of the baseline classifiers for each given input. In general, if n_1 and n_2 are the number of correctly classified samples by the two baseline components and n_{12} is the number of test samples that are correctly classified by both the baseline components, then the maximum achievable micro-average F-score by the proposed framework will be given by $\frac{n_1+n_2-n_{12}}{n}$ where n is the total number of test samples. We empirically make an

Table 5. Max achievable MicroF by combining baselines

Classifier	Max
CNN	95.49
BiLSTM	94.25
SVM	94.76
LR	93.87

Table 6. Performance obtained by neural ensembling of the baseline classifiers (using word in isolation and using contextual information)

Classifier	as	bn	en	hn	MacF	MicF
CNN	86.39	86.24	84.94	90.28	87.33	86.56
BiLSTM	86.09	85.89	84.55	86.26	86.17	85.61
SVM	86.46	86.74	89.94	90.04	87.53	86.59
LR	85.20	83.95	85.21	87.27	85.73	85.46

estimate of the maximum achievable performance by choosing the best predictions from each of the classifiers which is shown in Table 5.

5.1.4. Performance obtained using ensemble classification

Ensemble frameworks are commonly used frameworks to integrate information from different classification frameworks. This study, therefore, also uses a neural ensemble framework to combine the output of the baseline classifiers. Classification outputs (output probabilities) from the baseline classifiers are used as feature vectors which are fed to a fully connected hidden layer followed by a fully connected output layer. This performance is reported in Table 6. It is observed that the performance of the ensemble framework is very close to that obtained using the classifier considering contextual information with very marginal improvement.

5.1.5. Visualisation of the word embeddings

Before diving further into the results obtained using the different proposed strategies, we first investigate the characteristics of the pre-trained word embeddings that are used as implicit or explicit features in different classification set-ups. The word-level language classification strategies discussed in this paper are founded on the strong assumption that the word embeddings obtained reflect the language information of the words. To validate this assumption, a 2D projection of the word embeddings (obtained using t-SNE^b) is shown in Figure 3. It should be noted that the plot has been obtained only for words in the resolved set \mathcal{R} , since the languages for words in the unresolved set \mathcal{U} are not known. Figure 3 clearly shows that words in the same language are well clustered. Further, the plot in Figure 4 shows the average of the precision@k considering all words in \mathcal{R} . Given a query word from \mathcal{R} , the precision at k is obtained by calculating the ratio of the number of words that belong to the same language as the query word to the total number of words retrieved (k). The average of the precision at k is obtained by taking the average over all words in \mathcal{R} . It is observed that an average precision of 80% is obtained at $k = 1$, an average

^b<https://lvdmaaten.github.io/tsne/>

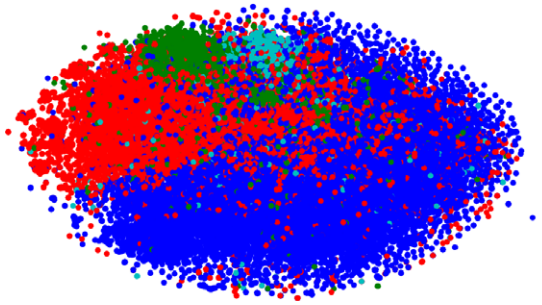


Figure 3. 2D projection of word2vec word embeddings.

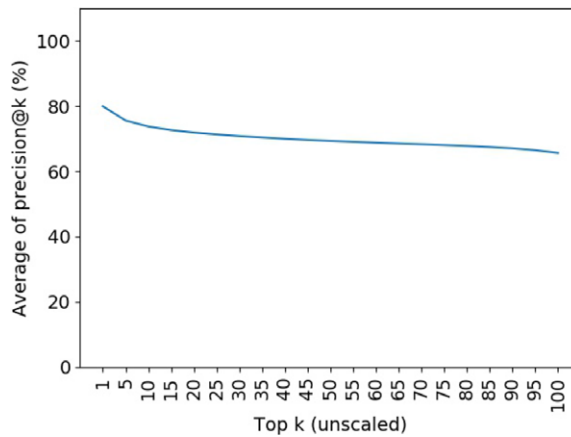


Figure 4. Average of precision@k for words in R .

precision greater than 70% is achieved for $k \leq 45$ and an average precision greater than 65% is achieved for $k \leq 100$. Therefore, combining the observations from Figures 3 and 4, it can be safely assumed that the pre-trained word embeddings capture the language information of the words.

5.2. Performance of the proposed framework

Table 7 shows the performance obtained using the proposed framework. Among the different baseline classification set-ups, using CNN as the baseline classifier shows the best performance. In comparison with the baseline classifiers, we find that the proposed framework exhibits improvement compared to the performance of the baseline classifiers. Further, using the proposed framework, it is seen that among the different features used to represent the input, the content-based features show the best performance, with the combination of all features performing marginally better than the text-based features alone. In the following discussion, we explore the performance of the proposed model using varying number of training samples and using different features.

5.2.1. Performance obtained with varying number of training samples

Figure 5 shows the performance of the proposed framework with varying number of training samples. Interestingly, the switch network is able to converge with a small dataset (87.5% for 10 samples to 90.1% for 750 number of samples). As observed in Figure 5, no significant change in

Table 7. Language-wise and macro- (MacF) and micro- (MicF) average F-scores obtained using the proposed framework under different classification set-ups and the relative performance with respect to the baseline components

Classifier	Features	as	bn	hn	en	MacF	MicF
CNN	Content	91.49	87.54	91.27	91.21	90.20(+3.10%)	90.49(+4.47%)
	Local neighbourhood distance	86.96	86.36	86.21	91.52	88.13(+0.74%)	87.44(+0.95%)
	Global neighbourhood	90.45	88.80	90.85	91.04	90.37(+3.48%)	90.47(+4.51%)
	Combination of all features	90.89	87.94	91.93	91.55	90.63(+3.60%)	90.98(+5.04%)
	Combination with feature dropout	91.09	89.05	91.72	90.97	90.79(+3.96%)	91.03(+5.16%)
BiLSTM	Content	89.68	86.98	89.43	87.02	88.48(+2.28%)	88.74(+3.54%)
	Local neighbourhood distance	86.19	85.93	84.17	86.91	86.36(−0.16%)	85.75(+0.05%)
	Global neighbourhood	88.41	83.61	90.42	86.73	87.61(+1.28%)	88.24(+2.96%)
	Combination of all features	90.00	87.12	89.76	87.05	88.69(+2.53%)	89.00(+3.79%)
	Combination with feature dropout	90.17	87.14	89.98	86.85	88.74(+2.58%)	89.10(+3.90%)
SVM	Content	89.16	84.16	89.89	90.22	88.44(+1.36%)	88.93(+2.80%)
	Local neighbourhood distance	86.28	84.83	85.05	90.30	86.84(−0.46%)	86.38(−0.13%)
	Global neighbourhood	88.57	83.88	89.63	90.08	88.20(+1.08%)	88.62(+2.45%)
	Combination of all features	88.91	83.93	90.12	89.15	88.22(+1.11%)	88.77(+2.62%)
	Combination with feature dropout	89.67	84.39	90.25	90.39	88.76(+1.73%)	89.27(+3.20%)
LR	Content	88.43	83.52	89.68	87.27	87.37(+1.32%)	88.00(+2.82%)
	Local neighbourhood distance	85.51	84.23	85.32	87.27	85.85(−0.44%)	85.58(+0.00%)
	Global neighbourhood	87.50	83.07	89.44	87.25	87.04(+0.93%)	87.60(+2.36%)
	Combination of all features	88.78	84.16	89.78	87.91	87.80(+1.82%)	88.35(+3.23%)
	Combination with feature dropout	89.13	84.38	89.26	87.94	87.81(+1.83%)	88.29(+3.16%)

performance is obtained on increasing the number of training samples. An analysis shows that the misclassifications are due to noisy word forms and noisy context. Therefore, it is assumed that further improvement in performance can be obtained only with the help of supervised information through annotated resources and knowledge bases.

5.2.2. Performance obtained using different features

Section 3.4 discusses the different ways used to represent the input before feeding it to the proposed model. This discussion highlights the following major observations.

- (1) **Content-based features show good performance across all set-ups.** The content features in the current experimental set-up are represented by the corresponding word embeddings. Therefore, the good performance of the content features can be attributed to the pre-trained word embeddings. When trained over a large corpus, the word embeddings are expected to capture the inherent similarities between words in the same language as well as dis-similarities between words in different languages. The word embedding of the

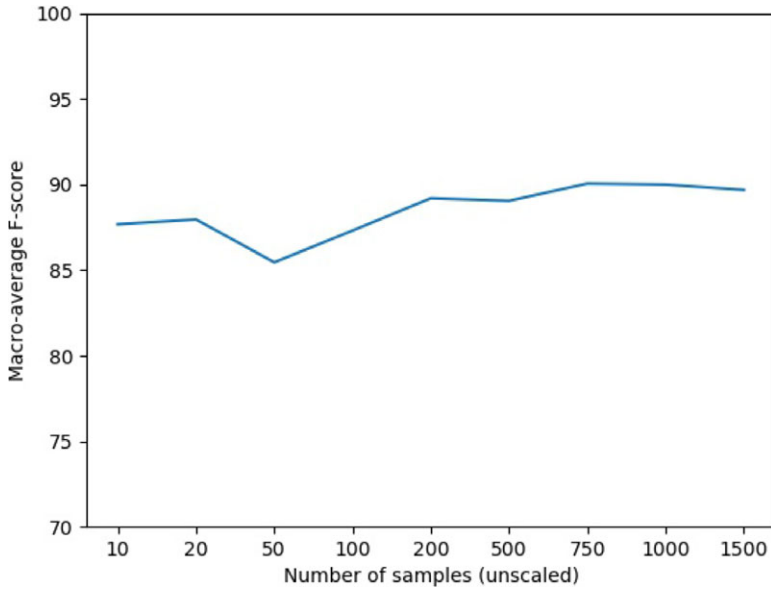


Figure 5. Performance obtained using the proposed model (content features) with varying number of training samples.

target word combined with that of the context serves as a representation of the word in a particular context. The performance obtained using words in \mathcal{R} as training samples shows that the word representations thus obtained are distinctive in nature in terms of language identification.

However, the word embeddings used have a few disadvantages. First, embedding of the same word in different senses are conflated into a single embedding. Therefore, occurrences of the same word in different languages is combined into a single embedding. Second, due to large number of spelling variations in social media data, word embeddings of infrequent word forms may not be reflective of their actual language characteristics.

- (2) **Neighbourhood-based features do not excel but exhibit performance that is comparable to the content-based features.** The neighbourhood-based features (the local neighbourhood and the global neighbourhood) represent finite structural information about the target input and its context in contrast to the content-based features that represent detailed semantic information. Although the neighbourhood-based features do not excel the content-based features in all set-ups, the performance is comparable to the content-based features. These observations convey two important advantages of the proposed neighbourhood-based features. First, the neighbourhood-based information can be used complementarily to the content-based information to counter the ambiguities resulting from content-based features. For example, when a word is embedded into a sentence of another language, the global neighbourhood-based features can be the correct indicator of the language of the word. Second, the finite-sized neighbourhood vectors are able to capture the same amount of information as the comparatively larger word embedding vectors thus leading to a reduction in the amount of computational resources required.

Further, considering that the neighbourhood features are computed based on the embeddings that are themselves generated from noisy user-generated text, it can be expected that with cleaner neighbourhood representations, there will be a further boost in performance.

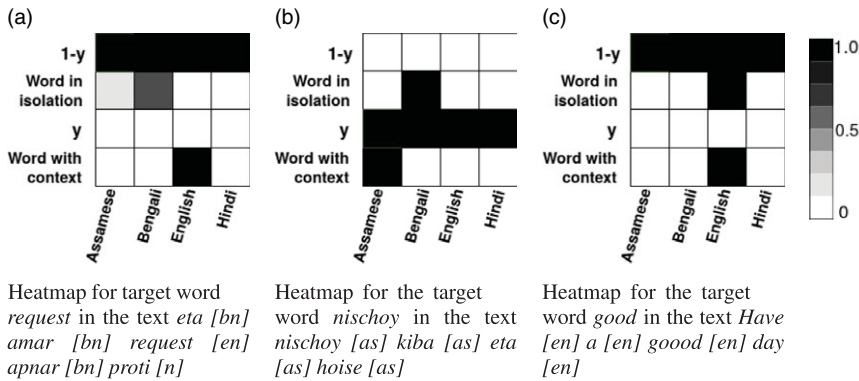


Figure 6. Figure showing the value of switch vector y for different input text.

(3) **Combination of all features.** Although the neighbourhood-based features (local neighbourhood and global neighbourhood) individually do not outperform the content-based features across all set-ups, in combination with the content-based features, they add to improvement compared to the performance obtained using content-based features only. However, the improvement is only minimal. This minimal improvement can be explained from the experimental observations where it is observed that the results obtained from the content-based features and the neighbourhood-based features are highly co-related. This is also true conceptually because the neighbourhood-based features are a structured representation of the content-based features. The structured information resolves some of the ambiguities that may be inflicted by the content features. Therefore, some of the misclassifications resulting from the content-based features are correctly classified by the neighbourhood-based features. However, in the attempt to extract structured features from the unstructured content-based features into a fixed-sized neighbourhood vector, there is also a loss of information. This is why the performance of the neighbourhood-based features individually is comparatively lower than the content-based features in certain set-ups.

However, as mentioned above, since the neighbourhood vectors used in this study are themselves obtained from word embedding vectors that are generated from noisy text, it can be expected that with cleaner representations, they should be able to outperform the content features.

5.3. Analysis of the switch layer

The objective of the proposed framework is to make an optimal choice between the baseline classifiers using a dynamic switching mechanism. The vector y discussed in Section 3.3 acts as the switch. To validate whether y actually acts as a switch, we analyse the values of y . Figure 6 shows a visualisation of the value of y for three different examples using a heatmap where the colour black corresponds to 1 and white corresponds to 0, and the intermediate values correspond to the intermediate shades.

Figure 6(a) is an example where the target word *request*, which is an English word, is embedded into a transliterated Bengali sentence *eta amar request apnar proti [this is my request to you]*. Therefore the class label obtained using the word in isolation should be given more priority over that obtained using the contextual information. We see that the weight associated with the output from the classifier using the contextual information, that is, the values in y approximate to 0, making the values in $1 - y$ approximately one satisfying our objective. Similarly, Figure 6(b) is an example where both the target word *nischoy [sure]* and the context *nischoy kiba eta hoise [definitely]*

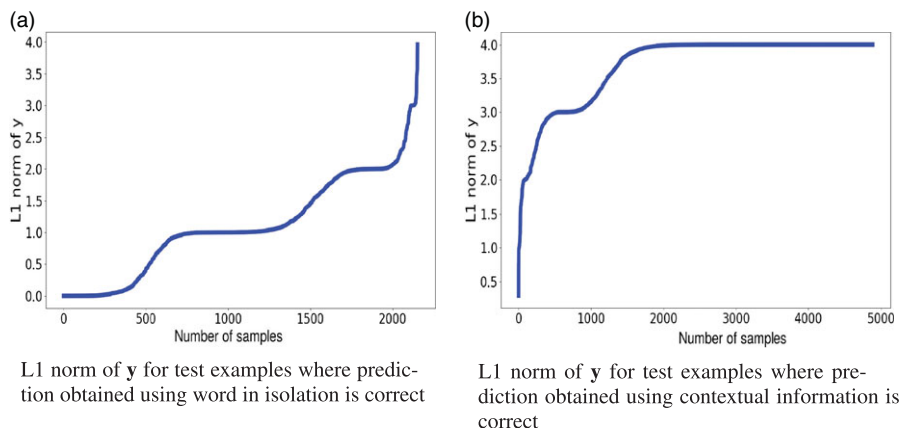


Figure 7. L1 norm of the switch vector y .

something must have happened] are in the same language, and therefore, the class label predicted by the classifier using the contextual information should be given more priority which is also validated from the values of y and $\mathbf{1} - y$ in the Figure. Figure 6(c) is an example where the class labels predicted using both the baseline classifiers are the same. Therefore, the switch vector y may be biased towards either prediction (word in isolation in this case).

The above illustration is for a few particular examples. In order to visualise the characteristic over a larger set, we analyse the L1 norm of y for the two possible cases: first, when the output should be guided by the word in isolation or *global semantic similarity* and second, when the output should be guided by the contextual information or *local contextual similarity*. This is shown in Figure 7 and is coherent with our assumption that when the switch should be towards the *global semantic similarity*, the L1 norm of y should be low and high when the switch should be towards the *local contextual similarity*. Figure 7 confirms our intuition.

5.4. Handling unseen and shared vocabulary

The test data considered in this study are not seen during either training or validation. Further, all words in the test data are words that belong to the *unresolved set* \mathcal{U} , that is, words that occur across sentences of multiple languages. These occurrences could be either due to similarities between languages or due to embedding or borrowing of words in sentences of languages other than the native languages of the words. As such, the maximum macro- and micro-average F-scores of 90.79% and 91.03%, respectively, achieved using the proposed model indicate the capability of the model in dealing with unseen and shared vocabulary.

6. Conclusion and future work

This study investigates the problem of word-level language identification for code-mixed social media text in a multilingual environment. The objective is to boost language identification performance by combining outputs from different baseline classifiers, each of which captures information that is non-complementary in nature. Results obtained indicate that the proposed framework is able to make the correct choice between the outputs of the baseline classifiers when one of the outputs is incorrect. The results also indicate the effectiveness of the proposed model in addressing unseen and shared vocabulary. Further, the proposed model uses minimum annotated resources and no external resources, making it a suitable framework for language identification in low-resource scenarios. An important observation from this study is

that non-textual features like neighbourhood-based features are as good as text-based features in capturing information necessary for language identification. One of the limitations of the neighbourhood-based features used in this study is that they are derived from the word embeddings that are noisy in nature. Therefore, in future, we would like to explore better non-textual features (neighbourhood-based/distance-based features) that can exceed the performance of the text-based features.

Even though the proposed approach shows improved performance compared to the baseline components, the performance is not equal to the projected best performance (Table 5). Analysis of the incorrect predictions shows that incorrect predictions are due to (i) noisy and infrequent word forms and (ii) noisy context. Therefore, it is assumed that further improvement in performance can only be achieved by incorporating information from external resources like dictionaries and knowledge bases. Therefore, in future, our efforts will be directed towards boosting the performance of the proposed framework further by incorporating information from external resources. Another area of exploration in this direction is the use of state-of-the-art embedding models like BERT (Devlin *et al.* (2019)) that support fine-tuning features based on the end classification task. Since training BERT models is expensive and requires large amounts of data, we do not consider it within the scope of the current work.

References

- Abainia K., Ouamour S. and Sayoud H. (2016) Effective language identification of forum texts based on statistical approaches. *Information Processing & Management* 52, 491–512.
- Banerjee S., Kuila A., Roy A., Naskar S.K., Rosso P. and Bandyopadhyay S. (2014) A hybrid approach for transliterated word-level language identification: Crf with post-processing heuristics. In *Proceedings of the 2014 Forum for Information Retrieval Evaluation*, Bangalore, India, pp. 54–59.
- Barman U., Das A., Wagner J. and Foster J. (2014) Code mixing: A challenge for language identification in the language of social media. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, Doha, Qatar, pp. 13–23.
- Bock Z. (2013) Cyber socialising: Emerging genres and registers of intimacy among young south african students. *Language Matters* 44, 68–91.
- Bullock B., GuzmÑn W., Serigos J., Sharath V. and Toribio A.J. (2018) Predicting the presence of a matrix language in code-switching. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, Melbourne, Australia, pp. 68–75.
- Carter S., Weerkamp W. and Tzagkias M. (2013) Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text. *Language Resources and Evaluation Journal* 47(1), 195–215.
- Cavnar W.B. and Trenkle J.M. (1994) N-gram-based text categorization. In *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, USA, pp. 161–175.
- Chandu K., Manzini T., Singh S. and Black A.W. (2018) Language informed modeling of code-switched text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, New Orleans, Louisiana, pp. 92–97.
- Chittaranjan G., Vyas Y., Bali K. and Choudhury M. (2014) Word-level language identification using CRF: Code-switching shared task report of MSR india system. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, Doha, Qatar, pp. 73–79.
- Das A. and Gamback B. (2014) Identifying languages at the word level in code-mixed Indian social media text. In *Proceedings of the International Conference on Natural Language Processing*, Goa, India, pp. 378–387.
- Das S.D., Mandal S. and Das D. (2019). Language identification of Bengali-English code-mixed data using character & phonetic based LSTM models. In *Proceedings of the Forum for Information Retrieval Evaluation*, Kolkata, India, pp. 60–64.
- Devlin J., Chang M.-W., Lee K. and Toutanova K. (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, Minneapolis, Minnesota, pp. 4171–4186.
- Garg A., Gupta V. and Jindal M. (2014) A survey of language identification techniques and applications. *Journal of Emerging Technologies in Web Intelligence* 6, 388–400.
- Gella S., Bali K. and Choudhury M. (2014) ye word kis lang ka hai bhai? Testing the limits of word level language identification. In *Proceedings of the International Conference on Natural Language Processing*, Goa, India, pp. 368–377.
- Gundapu S. and Mamidi R. (2018) Word level language identification in english telugu code mixed data. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation*, Hong Kong, pp. 180–186.

- Jaech A., Mulcaire G., Hathi S., Ostendorf M. and Smith N.A.** (2016) Hierarchical character-word models for language identification. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, Austin, TX, pp. 84–93.
- Jauhainen T.S., Lui M., Zampieri M., Baldwin T. and Lindén K.** (2019) Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research* **65**, 675–782.
- Jurgens D., Tsvetkov Y. and Jurafsky D.** (2017) Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, pp. 51–57.
- King B. and Abney S.** (2013) Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, pp. 1110–1119.
- Mager M., Cetinoglu O. and Kann K.** (2019) Subword-level language identification for intra-word code-switching. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, pp. 2005–2011.
- Mandal S. and Singh A.K.** (2018) Language identification in code-mixed data using multichannel neural networks and context capture. In *Proceedings of The Fourth Workshop on Noisy User-generated Text*, Brussels, Belgium, pp. 116–120.
- 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*, Melbourne, Australia.
- Miyamoto Y. and Cho K.** (2016) Gated word-character recurrent language model. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Texas, USA, pp. 1992–1997.
- Molina G., Rey-Villamizar N., Solorio T., Alghamdi F., Ghoneim M., Hawwari A. and Diab M.** (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*, Texas, USA, pp. 40–49.
- Nguyen D. and Cornips L.** (2016) Automatic detection of intra-word code-switching. In *Proceedings of the Fourteenth SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Berlin, Germany, pp. 82–86.
- Nguyen D. and Doğruöz A.S.** (2013) Word level language identification in online multilingual communication. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Washington, USA, pp. 857–862.
- Papalexakis E., Nguyen D. and Doğruöz A.S.** (2014) Predicting code-switching in multilingual communication for immigrant communities. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, Doha, Qatar, pp. 42–50.
- Patro J., Samanta B., Singh S., Basu A., Mukherjee P., Choudhury M. and Mukherjee A.** (2017) All that is English may be Hindi: Enhancing language identification through automatic ranking of the likeliness of word borrowing in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 2264–2274.
- Piergallini M., Shirvani R., Gautam G.S. and Chouikha, M.** (2016) Word-level language identification and predicting codeswitching points in swahili-english language data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, Texas, USA, pp. 21–29.
- Rei M., Crichton G. and Pysalo S.** (2016). Attending to characters in neural sequence labeling models. In *Proceedings of the 26th International Conference on Computational Linguistics*, Osaka, Japan, pp. 309–318.
- Rijhwani S., Sequiera R., Choudhury M., Bali K. and Maddila C.S.** (2017) Estimating code-switching on twitter with a novel generalized word-level language detection technique. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, pp. 1971–1982.
- Rudra K., Sharma A., Bali K., Choudhury M. and Ganguly N.** (2019) Identifying and analyzing different aspects of English-Hindi code-switching in Twitter. *ACM Transactions on Asian and Low-Resource Language Information Processing* **18**.
- 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*, Texas, USA, pp. 50–59.
- Sarma N., Sanasam R. and Goswami D.** (2019) Influence of social conversational features on language identification in highly multilingual online conversations. *Information Processing & Management* **56**, 151–166.
- Sarma N., Singh S.R. and Goswami D.** (2018) Word level language identification in Assamese-Bengali-Hindi-English code-mixed social media text. In *Proceedings of the 2018 International Conference on Asian Language Processing (IALP)*, Bandung, Indonesia, pp. 261–266.
- Sikdar U.K. and Gambäck, B.** (2016) Language identification in code-switched text using conditional random fields and babelnet. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, Texas, USA, pp. 127–131.
- Singh K., Sen I. and Kumaraguru P.** (2018) A Twitter corpus for Hindi-English code mixed pos tagging. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, Melbourne, Australia, pp. 12–17.

- Solorio T., Blair E., Maharjan S., Bethard S., Diab M., Ghoneim M., Hawwari A., AlGhamdi F., Hirschberg J. and Chang A.** (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*, Doha, Qatar, pp. 62–72.
- Volkova S., Ranshous S. and Phillips L.** (2018) Predicting foreign language usage from English-only social media posts. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana, pp. 608–614.
- Vyas Y., Gella S., Sharma J., Bali K. and Choudhury M.** (2014) Pos tagging of English-Hindi code-mixed social media content. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 974–979.
- Wang P., Bojja N. and Kannan S.** (2015) A language detection system for short chats in mobile games. In *Proceedings of the International Workshop on Natural Language Processing for Social Media*, Denver, Colorado, pp. 20–28.
- Xia M.X.** (2016) Codeswitching language identification using subword information enriched word vectors. In *Proceedings of The Second Workshop on Computational Approaches to Code Switching*, Texas, USA, pp. 132–136.
- Yang X. and Liang W.** (2010) An n-gram-and-wikipedia joint approach to natural language identification. In *Proceedings of the International Universal Communication Symposium*, Beijing, China, pp. 332–339.
- Yip, V. and Matthews, S.** 2016. Code-mixing and mixed verbs in Cantonese-English bilingual children: Input and innovation. *Languages, MDPI*, 1(1):4–18.
- Zhang Y., Riesa J., Gillick D., Bakalov A., Baldrige J. and Weiss D.** (2018) A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 328–337.
- Zubiaga A., San Vicente I., Gamallo P., Pichel J.R., Alegria I., Aranberri N., Ezeiza A. and Fresno V.** (2016) Tweetlid: A benchmark for tweet language identification. *Language Resources and Evaluation Journal* 50, 729–766.