IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 March 2023, accepted 2 May 2023, date of publication 18 May 2023, date of current version 1 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3277787

RESEARCH ARTICLE

Data-Augmentation for Bangla-English Code-Mixed Sentiment Analysis: Enhancing Cross Linguistic Contextual Understanding

MOHAMMAD TAREQ[1], MD. FOKHRUL ISLAM[2], SWAKSHAR DEB[2], SEJUTI RAHMAN[2], AND ABDULLAH AL MAHMUD[3]

[1]Department of Accounting and Information Systems, University of Dhaka, Dhaka 1000, Bangladesh
[2]Department of Robotics and Mechatronics Engineering, University of Dhaka, Dhaka 1000, Bangladesh
[3]Department of Banking and Insurance, University of Dhaka, Dhaka 1000, Bangladesh

Corresponding author: Sejuti Rahman (sejuti.rahman@du.ac.bd)

This research was supported by the Centre for Advanced Research in Strategic Human Resource Management (CARSHRM), University of Dhaka.

ABSTRACT In today's digital world, automated sentiment analysis from online reviews can contribute to a wide variety of decision-making processes. One example is examining typical perceptions of a product based on customer feedbacks to have a better understanding of consumer expectations, which can help enhance everything from customer service to product offerings. Online review comments, on the other hand, frequently mix different languages, use non-native scripts and do not adhere to strict grammar norms. For a low-resource language like Bangla, the lack of annotated code-mixed data makes automated sentiment analysis more challenging. To address this, we collect online reviews of different products and construct an annotated Bangla-English code mix (BE-CM) dataset (Dataset and other resources are available at https://github.com/fokhruli/CM-seti-anlysis). On our sentiment corpus, we also compare several alternative models from the existing literature. We present a simple but effective data augmentation method that can be utilized with existing word embedding algorithms without the need for a parallel corpus to improve cross-lingual contextual understanding. Our experimental results suggest that training word embedding models (e.g., Word2vec, FastText) with our data augmentation strategy can help the model in capturing the cross-lingual relationship for code-mixed sentences, thereby improving the overall performance of existing classifiers in both supervised learning and zero-shot cross-lingual adaptability. With extensive experimentations, we found that XGBoost with Fasttext embedding trained on our proposed data augmentation method outperforms other alternative models in automated sentiment analysis on code-mixed Bangla-English dataset, with a weighted F1 score of 87%.

INDEX TERMS Code mixed, sentiment analysis, Bangla-English corpus, bi-lingual, zero-shot learning.

I. INTRODUCTION

Customer sentiment analysis has piqued the interest of the business community, who wants to learn what customers think about their products or services [1], [2], [3]. It has become popular in many businesses, including mobile banking, online retail, and restaurants, among others. It is apparent that English, being considered as a ''universal language'' is often chosen by different nationalities to communicate over the internet to express their positive or negative feelings about a product. When expressing opinions or thoughts, people in bi or multilingual communities are more inclined to use their local language in addition to English. As a result, it encourages ''code-mixing'', which is the mingling of several languages inside a sentence. This code mixing is a common phenomenon in multilingual societies such as Bangladesh and India [4], [5]. Bangla is the fifth most spoken native language in the world, with approximately 300 million native

The associate editor coordinating the review of this manuscript and approving it for publication was Alicia Fornés.

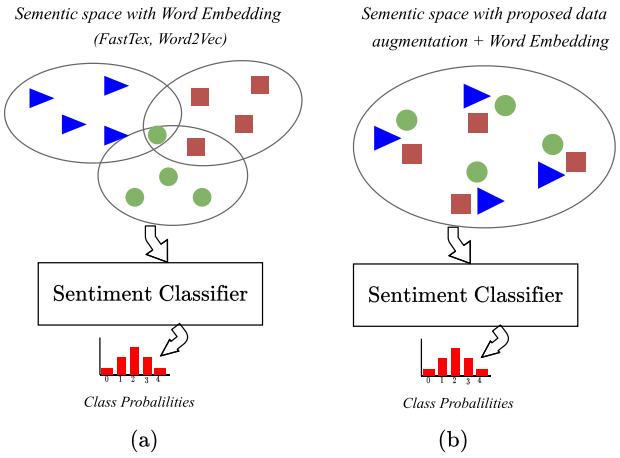

FIGURE 1. Distinct languages are represented by different colors (blue: English, brown: Bangla, green: transliterated Bangla) in a shared semantic space for CM sentiment classification. (a) Previous studies have used existing monolingual word embeddings for CM sentiment analysis, and therefore, words from different languages cannot be related. (b) When the proposed data augmentation is paired with existing word embeddings, cross-lingual understanding is developed, which improves CM sentiment classification performance.

speakers. This Indo-Aryan language is spoken not just in Bangladesh but also in some parts of India - West Bengal, Tripura, and Assam. Apart from the ''code-mixing'' trend, most internet users would rather type in Roman characters or use phonetic typing in their first language than utilize Unicode. ''Transliteration'' is the term for this phenomenon. Despite the fact that mixing languages and using phonetic typing is considered a sub-standard usage of language in formal communications, it has become a normal and integral aspect of communication in the much less formal context of social media (e.g., product reviews). In light of the growing number of people from different countries who use social media, this study focuses on sentiment analysis in Code Mixed (CM) Bangla-English, which has very little annotated data.

In this paper, we develop a corpus of CM sentiments, BE-CM dataset, with 18,074 samples and provide a novel data augmentation approach for performing sentiment analysis that is capable of handling the combination of languages at different levels in CM sentences. We identify the following challenges associated with sentiment analysis using CM sentences:

(1) Due to the fact that phonetic typing (transliteration) does not adhere to standard spelling conventions, different individuals may type the same word differently, *bhalo* or *bala* or *valo* (Good), all of which express a similar contextual meaning. ***(2)*** The transliterated Bangla does not follow any grammatical rules, rendering part of speech tagging and lexicon-based approaches [6], [7] ineffective. ***(3)*** Prior works on code-mixed sentiment analysis have utilized a variety of word embedding techniques, including word2vec[8], glove[9], and fasttex [10]. These monolingual embeddings fail to capture the cross lingual relationship among similar words across different languages (See Figure 1(a)). ***(4)*** While some research has been conducted on developing approaches for extracting sentiment from monolingual English or Bangla corpora, there is a dearth of studies on code-mixed Bangla-English sentiment analysis due to a lack of available corpora for supervised learning. As a result, this area of literature is underserved. This work makes an attempt to resolve the aforementioned issues.

To overcome the scarcity of training data, we construct a gold standard Bangla-English code-mixed sentiment dataset. This dataset comprises a total of 18,074 code-mixed sentences with annotated ground truth, 14,459 for training and 3,615 for testing. The corpus is benchmarked and the results are analyzed using several machine learning and deep learning models. To aid in the research for resource-constrained code mixing in the Bangla-English language, we also examine zero shot cross-lingual transfer[11]. The basic concept is to train a classifier on resource-rich monolingual English sentiments and then test it on low-resource languages such as Bangla-English. Additionally, we presented a simple yet effective data augmentation method that can be used in conjunction with the current monolingual word embedding models to improve cross-linguistic contextual understanding for code-mixed sentiment analysis (See Figure 1(b)). Furthermore, this augmentation approach eliminates the requirement for a parallel corpus, which is wasteful given the constantly changing nature of code-mixed sentences with multiple spelling and word variations. Our proposed data-augmentation consists of the following steps: ***(i)*** To begin, we extract each sentence from the review. ***(ii)*** Secondly, within each sentence, we select words based on a sampling rate parameter, which determines the frequency of the selected words. ***(iii)*** Following that, using a dictionary, the selected words are transformed to their monolingual counterparts. ***(iv)*** Along with the original reviews, the augmented reviews are then utilized to train an existing word embedding model.

Considering the importance of extracting sentiments from millions of social media texts and paucity of research in Bangla-English code-mixed language, this work makes the following contributions:

- We construct the BE-CM dataset, the first large-scale code-mixed Bangla-English annotated dataset for sentiment analysis.
- We benchmark our CM dataset for sentiment classification using logistic regression[12], support vector machine[13], decision tree[14], 1DConv-LSTM, XGBoost[15] and various BERT models[16], [17], [18].
- We propose a simple and effective data augmentation method for capturing cross-lingual relationships without the requirement for a parallel corpus.

II. RELATED WORKS

A. SENTIMENT ANALYSIS ON MONOLINGUAL CORPUS

As mentioned earlier, sentiment analysis gives useful insights into client opinions regarding particular products, [2], [19],

apps [20], and social media [21]. Many works have been done on monolingual sentiment analysis in English, including [2], [21], [22], [23], [24] with handcrafted features. For example, the work in [21] did sentiment analysis on Twitter data. They introduced POS-specific senti-features to predict the sentiment of comments. Reference [2] also conducted sentiment analysis on product review data from ''amazon.com''. The sentiment categorization was performed at the sentence and review levels and yielded satisfactory results in both cases. Often, the opinion is not explicitly articulated. In these cases, the traditional approach fails. Reference [24] tackled this problem in their gap analysis of customer reviews of service quality. They collected English reviews from a variety of online sources and built a service-feature hierarchy. They provided customer perception scores (CPS) and customer expectation scores (CES) based on calculated features. Recently, deep learning-based methods have been increasingly popular in sentiment analysis [25], [26] because of their superior performance. Sun and Wang[25], for example, offered a sentiment analysis approach based on deep learning. It employs the Regional CNN (RCNN) to preserve the temporal relationship between sentences while collecting extra semantic connections between words. They overcome the issue that the previous model has fewer connections between sentences and less semantic information between words when they tackle the aspect-based sentiment analysis task. He et al.[26] developed a word, part of speech pairs-CNN (WP-CNN) model to improve the representation quality of input text. It takes into consideration the characteristics of parts of speech in order to enhance word embedding representation.

Sentiment analysis on monolingual Bangla datasets has gained popularity in the NLP community. Earlier works [27], [28], [29], [30] used a variety of machine learning algorithms to classify sentiment. The datasets were compiled from a variety of sources, including Twitter [5], [27], [31], [32], Facebook[28], [33], and newspaper[29]. For example, Chowdhury et al. [27], Ali et al.[34], Islam et al. [28], Mahtab et al. [30] and Ghosal et al. [29] created their own datasets from various sources. These approaches entail a number of preprocessing steps, such as pos-tagging, eliminating punctuation, and deleting stopwords, which impede the end-to-end learning process. Moreover, the works in [27], [28], and [29] have limited capacity to understand varying semantic relations within words. To tackle this problem, Mandal et al. [5] proposed a hybrid model combining the Stochastic Gradient Descent Classifier and a rule-based method. On the other hand, some studies [32], [33] also incorporate deep learning based techniques to further improve the sentiment classification. Sarkar and Bhowmick [32] accumulate vectorization based CNN method for this purpose. In addition, Hassan et al. [33] proposed an LSTM-based method to capture the sequential information in monolingual sentences. Apart from that, Ali et al. [34] presented a lexicon-based corpus that relies solely on the polarity of words and can be used to understand the overall sentiment of sentences. However these datasets cannot be used for sentiment analysis on code-mixed setups which is common in multilingual communities.

B. SENTIMENT ANALYSIS ON CODE-MIXED CORPUS

Due to the increased use of social media, communication, and opinions, code-mixed sentiment analysis has recently gained popularity in Hindi-English [35], [36], [37], Malaylam-Tamil-English [38], Bangla-English [39], German-English[40] and others. These studies serve a range of objectives, including social media analysis [35], [41], cyber bulling [36], [37], [42], product/restaurant reviews [43] and others[44]. Earlier efforts, such as [36], [40], [42], [43], and [45], employ a variety of preprocessing procedures (such as pos tagging, stemming, and tokenization) before feeding the review data into a machine learning classifier. Nowadays, researchers do this job using techniques based on deep learning [35], [36], [37], [41], [46]. For example, Singh et al. [35] developed a transfer learning based LSTM method to classify sentiment analysis for Hindi-English code-mixed tweets. They did not, however, consider misspellings and word variations that convey the same meaning in an informal setting such as social media. To address this, [47] took the spelling of the texts into account as well, by annotating the correct spelling, even in transliterated words. They applied language identification, normalization, and POS tagging algorithms sequentially in three different experiments. To study the effect of word embeddings, Pratapa et al. [46] compared three bilingual word embedding approaches: bilingual correlation based embeddings [48], bilingual compositional model [49] and bilingual Skip-gram [50], to perform code-mixed sentiment analysis and part-of-speech tagging. They found that the bilingual embeddings do not perform well since code-mixed text contains particular semantic and syntactic structures that do not occur in the respective monolingual corpora. The majority of prior research on cross-lingual sentiment models has relied on translation systems [51] or cross-lingual signals in other forms, such as parallel corpora [52]. However, because we are working with code-mixed (and transliterated) Bangla-English data, parallel corpora and language translators are scarce. Additionally, due to the ever-changing nature of social media content and the multiplicity of spelling alternatives, data-intensive approaches based on parallel corpora will be rendered outdated. Apart from that, one of the significant obstacles in this field of study is a lack of publicly available training data. However, according to the current literature, only Mandal et al. [5] collected a code-mixed Bangla-English corpus with 5000 samples for sentiment analysis. Therefore, this work constructed the BE-CM dataset utilizing correct annotation processes in order to accomplish code-mixed text identification and classification in Bangla with a considerable amount of training data.

C. CROSS-LINGUAL TRANSFER

Word embeddings are a sort of word representation that allows words with similar meanings to have similar

representations. There are several popular embedding algorithms for monolingual word embeddings, including Skip-gram with negative sampling [8], Continuous Bag-of-Words (CBOW) [8], Global Vectors (GloVe) [9] and Fasttext [10]. They each have a unique method of learning. For instance, the skip-gram method [8] learns surrounding word embeddings based on the context of the central word. On the other hand, CBOW predicts the center word jointly using all context words. GloVe [9] enables us to learn word representations via matrix factorization. It minimizes the difference between the dot product of a target word's embeddings and the context word's embeddings. These approaches, however, do not account for misspellings. However, in a code-mixed environment, it is common to come across misspelled terms, particularly in various social media posts or product evaluations. To address this, Fasttext [10] uses sub-word segmentation models using WordPiece and GloVe to train subword embeddings. However, these methods are designed for monolingual text, and one of the main objectives of this work is to understand cross-lingual relationships. To address this problem, Ruder et al. [53] looked at several methods [54], [55], [56] for learning cross-lingual word embeddings, including joint training and post-training mappings of monolingual embeddings. Xing et al.[57], Lample et al.[58], and Chen and Cardie [59] recommended aligning multilingual pre-trained monolingual word embeddings into a common semantic space using pre-trained monolingual word embeddings. Our work is part of a recent line of research on cross-lingual contextual understanding [16], [60], [61], [62], [63], [64], that employs masked language modeling or other auxiliary pre-training tasks to encourage closer representation in source and target language space. We propose a simple yet effective data augmentation method for dynamically generating code-switching data for training, which implicitly encourages the model to align similar phrases in romanized Bangla and English into the same space. Moreover, the augmentation technique we propose can also be utilized to produce synthetic data for low-resource languages in machine translation(MT) task[65], [66]. It has the benefit of repeating rare words by adding them to the dictionary and reducing noise in the source-to-target synthetic data conversion by utilizing different sampling rates.

III. CORPUS CREATION AND ANNOTATION

One of the primary challenges in sentiment analysis is the lack of publicly available code-mixed datasets. Researchers have published findings in the literature ([67], [68]) utilizing regional language datasets such as Hindi-English and Tamil-English, but no work on appropriate Bangla-English code-mixed sentiment analysis has been done to our knowledge. So, we have made a well-annotated dataset for the study of Bangla-English code mixing that we will make public soon.

We intend to create a reasonably sized code-mixed corpus with sentences containing well-defined feelings that will be useful for future research. We extracted around 970,852 reviews from Google Playstore comments using the *google-play-scraper*.[1] Many of them had phrases entirely in English or entirely in Bangla. So we used the *langdetect library*[2] to filter out non-code-mixed reviews based on language identification at the comment level. Finally, we had 18,074 Bangla-English reviews.

We observed intra- and inter-sentiment switching for CM sentences in our corpus containing translitarated Bangla-English sentiment. Most of the comments were written in transliterated Bangla script along with English code switching in between them. We illustrate these examples in Table 1.

TABLE 1. Example texts of our dataset.

Text	Label	Remarks
Onek din dore eta bebohar korteci khub valo kaj korche *I was using it for long time. It is working fine.*	Pos.	Bangla words written in Roman script with no English switch.
Aj sokal theme try korci. But doesn't work *I am trying this from morning but it does not work.*	Neg.	Intra-sentential switch
Bad apps ... prement dei na ... fake appps ... report this app *Bad apps...they do not give payment...fake apps...report this app.*	Neg.	Inter-sentential switch between clauses.

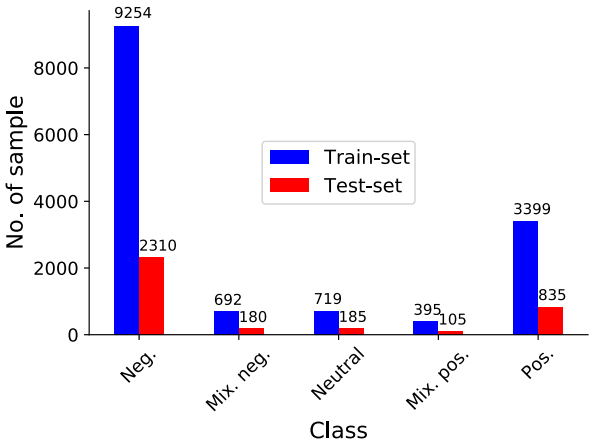

FIGURE 2. The total number of samples is distributed over five classes in the BE-CM dataset.

A. DATA COLLECTION

The dataset contains user comments for different apps (only used by Bangla-speaking people) from playstore. Using a web scrapping tool, we collect high-quality Bangla-English code-mixed data from the Google Play Store. A total of approximately 970,852 samples were collected. The reviews are in Bangla, English, Romanized Bangla. As we are only interested in reviews written in code-mixed Bangla-English

[1] https://pypi.org/project/google-play-scraper/
[2] https://pypi.org/project/langdetect/

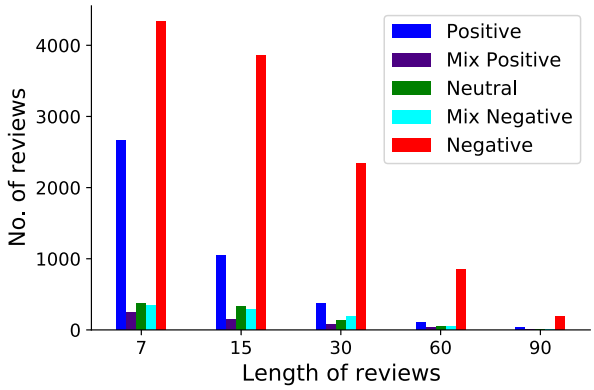

FIGURE 3. Number of sample vs length of reviews. We observed most reviews have 8-20 tokens.

TABLE 2. Corpus statistics of BE-CM dataset.

Parameter	
Total number of words	253,601
Number of reviews	18,074
Average number of words per post	14
Average number of sentences per post	1-2

comments. So, we excluded the monolingual English comments. We utilized a language detector to identify Bangla comments. After removing the monolingual English comments, the corpus contains around 18,074 thousand reviews.

B. CORPUS STATISTICS

We show the class and review length distributions of our collected dataset in Figure 2, 3 and corpus statistics in Table 2. This dataset is highly imbalanced, with the majority of sentiment being either positive (4,234) or negative (11,564). Additionally, we have 872 mixed negative reviews, 904 neutral reviews, and 500 mixed positive reviews. The entire dataset of 18,074 reviews was shuffled and split into two parts: 14,459 reviews for training and 3,615 for testing.

C. ANNOTATION SETUP

The corpus contains five types of sentiments as follows:

- **Positive state (Pos.):** The reviewer explicitly gives the clue that the comment is in a positive state, such as satisfied, happy, and admiring.
- **Negative state (Neg.):** There is an explicit clue in the text suggesting that the speaker is in a negative state, such as angry, sad, anxious, or violent.
- **Mixed positive (Mix. Pos.):** There is an explicit or implicit clue in the text suggesting that the speaker is experiencing both positive and negative feelings but with a bias towards the positive sentiment, such as forgiving.
- **Mixed negative (Mix. Neg.):** There is an explicit or implicit clue in the text that suggests that the speaker is feeling both positive and negative feelings, but with a preference for the negative.
- **Neutral:** There is no explicit or implicit indication of the reviewer's emotional state.

We anonymize user identities to protect each commentator's privacy when we collect data from the Google Play Store. We noticed several anomalies in the remarks, including the request for assistance and the inclusion of the reviewer's name. These comments are removed from the corpus.

D. DATA ANNOTATION

This data set was annotated by a pool of annotators representing a variety of genders and ages. It should be noted that the volunteer annotators' personal information (e.g., gender, education, medium of schooling) was acquired in order to understand more about them. The annotators were informed right away that their data would be logged and that they might opt out at any moment during the annotation process. The annotators should willingly give their consent to be recorded.

While collecting data from the Google Play Store, we noticed several inconsistencies in customer-provided labels (See Table 3), which were caused by a lack of a standardized procedure and individual subjectivity. As a result, it is critical to adhere to specific guidelines in order to ensure high-quality annotation and a deeper understanding of the dataset [69]. In a few instances in the literature, annotators are asked to assign a label based on their viewpoints only [70]. It is, however, risky, as individual interpretations and perceptions differ significantly. To prevent these issues, we delete the initial rating assigned by users and suggest annotators to adhere to the procedure provided in Figure 4 for annotation. Additionally, we used Google Forms to collect the email addresses of annotators, which we used to ensure that each annotator may only label a sentence once.

To decide the initial label, an annotator must first assess whether or not a text comprises a review. The text is removed if it is not a review. If this is a review, the annotator must assign an appropriate label (positive, negative, neutral, mixed positive, or mixed negative) according to the guideline (See Figure 4). Two people are responsible for annotating each review. We keep the provided label in its present form if both of them agree on it. A third annotator is assigned in the event of a dispute. If none of the three annotators can agree, the review is annotated by two more annotators who get the scope for a discussion. If the disagreement persists, we disregard this comment.

E. ANNOTATION QUALITY

To assess the annotations' validity and quality, we calculated the inter-annotator agreement. The Cohen's kappa coefficient [71] is used to determine the degree of agreement amongst annotators (Eq. 1).

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{1}$$

TABLE 3. Examples of divergence in annotations. The labels label-1 and label-2 indicate the rating given by the customer and our annotators respectively. The label 'Flag' indicates that the feeling stated does not fit into any of the predefined classes.

Text	Label-1	Label-2
Apatato 1 star dilam, payment palay 5 star dabo (*Currently I am giving 1 star, I will give 5 star after I receive the payment.*)	Neg.	Flag
Fake apps, baje ekta app baniya tokacce (*Fake app, this is a bad app that always charges more money.*)	Pos.	Neg.
Bashe valo na apps taka maira dai (*Not so good, takes more money than necessary.*)	Neg.	Pos.
Majhe slow hoye jay bt onk valo useful (*It gets slow sometimes but very useful app*)	Pos.	Mix. Pos.
50 takar bole 25 taka dicche...faltu (*They were supposed to pay 50 taka but only paid 25... disgusting*	Neutral	Neg.

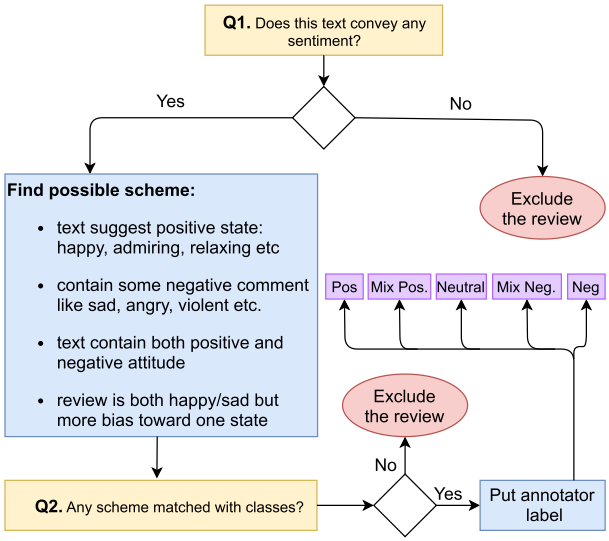

FIGURE 4. Guidelines for data annotation.

Here, p_o and p_e denote, respectively, the relative observed degree of agreement among annotators and the hypothetical probability of random agreement. The term "relative observed agreement" refers to the sum of all ratings agreed upon by annotators. If all annotators agree entirely, then $\kappa = 1$. And when there is no agreement amongst annotators, $\kappa = 0$. We reach a level of agreement of 0.76, which indicates substantial agreement.

IV. METHOD

A. PROBLEM FORMULATION

Assume that w_i is the ith review and y_i represents the associated sentiment. We represent the collection of all the corresponding comments and ratings as $\mathcal{D} = \{(w_1, y_1), \ldots, (w_n, y_n)\}$. The stop-wards, punctuation, and numerals are eliminated from those comments since they do not convey any useful information. After the preprocessing step, we perform our proposed code switching data augmentation as described in Sec IV-D to train the word embedding model. Note that, we only use this augmented sentences to train the embedding model. Once the embedding model is trained on the augmented sentences, we use it to transform each word, w_i, in the original corpus to a corresponding vector, x_i, in the embedding space, $x_i \in \mathbb{R}^d$. These word vectors are then feed to a classifier as an input. The classifier take this d-dimensional vectors as input and predicts a probability distribution that corresponds to the sentiment score. Finally, we make our final prediction based on the class with the greatest likelihood score.

TABLE 4. Spelling variations of translittered Bangla words in our Bangla-English code-mixed dataset.

Word	Meaning	Spelling variations
Shundor	Beautiful	Shundar Sundor, Sundhor
Bhalo	Good	Valo, Balo, Vhalo
Valobasa	Love	Vhalobasa, Valobassa, Valovasa
Bhul	Wrong	Bul, Vul, Vhul, Vhull

B. HANDLING CODE-MIXED WORD VARIATIONS

Transliteration languages that use a phonetic script (such as Bangla) results in word variations depending on the user(See Table 4). Those variants are all refer to the same term, with a similar functionality and context. The Word2Vec [8] model is trained to predict words that appear in their context, based on the distributional hypothesis [72]. The goal of the this method is to maximize the log-likelihood of the context words given a large training corpus. However, they ignore word internal structure by employing a separate vector representation for each word. To address this word variations, the Fasttext [10] enrich the word vectors with subword level information.

Fasttext [10] give the representation of a word in the semantic space based on both the context and subword information. For example, consider the word *where* with n-gram = 3. It will be represented by the character n-grams:

<wh, whe, her, ere, re>

and the special sequence

<where>.

Note that the sequence <her>, corresponding to the word *her* is different from the tri-gram her from the word where. In practice, fasttext extract all the n-grams for n greater or equal to 3 and smaller or equal to 6.

Suppose we have a dictionary of n-grams of size $\mathcal{G}$. Given a word w, $\mathcal{G}_w \subset \{1, 2, \ldots, N\}$ represents the set of n-grams appear in the word w, where N is the maximum n-gram size. Fasttext model learns a vector representation for each n-gram in $\mathcal{G}_w$, and represents the word, w, by the sum of the vector representations of its n-grams. This simple model allows fasttext to share the representations across words, thus allowing to learn reliable representations for word variations.

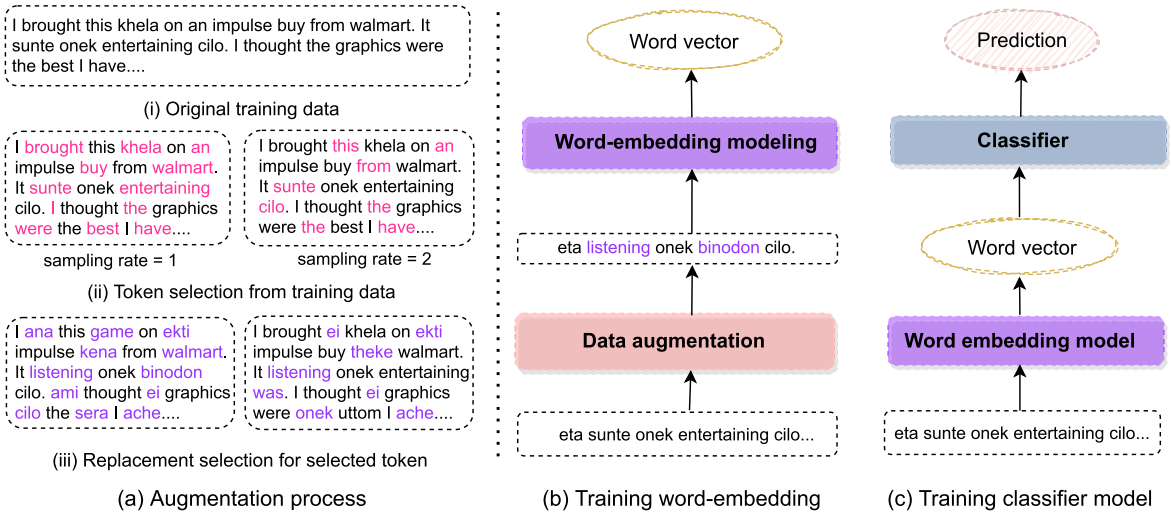

FIGURE 5. **(a) Proposed data augmentation process with multiple sampling rates. For simplicity, we only showed sampling rate 1 and 2 in the above diagram. (b) Illustration of word embedding training process. We augment input data with several sampling rate. (c) Training the classifier using learned word embedding.**

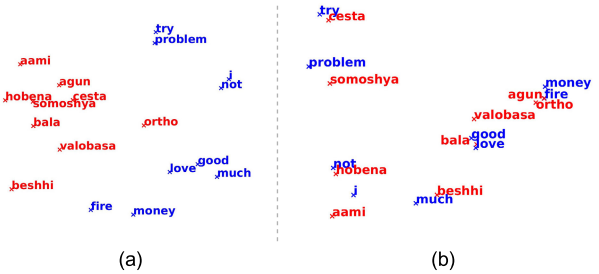

FIGURE 6. **Principal component analysis (PCA) of words in embedding space shown as a two-dimensional vector. Blue color indicates English words, whereas red color denotes the Bangla-counterpart. (a) Vector representation using Fasttext, in which equivalent multilingual words are separated by a wide distance due to the omission of cross-lingual connections. (b) Vector representation of the same multilingual words utilizing our suggested technique (Fasttext+data augmentation) where the similar words are grouped together.**

In aforementioned example, the word `where` not only learns a word vector for itself but also generates word vector representations for its tri-grams(`<wh, whe, her, ere, re>`). When fastText encounters an out-of-bag vocabulary word, it separates this into its n-grams and aggregates the corresponding vectors to calculate the word vector. This solves the problem of out-of-bag vocabulary words.

Algorithm 1 Proposed Data Augmentation Framework

Input: Given training data: $S = \{s^{(n)}\}_{n=1}^{N}$; a bilingual dictionary: *dict*; temporary corpus: $\mathbf{t} \leftarrow \emptyset$; *sampling rates*

Output: Augmented dataset: Φ.

for $n \leftarrow 1 \ldots N$ **do**
  $i, \alpha \leftarrow 0$;
  for r *in sampling rates* **do**
    while $s_i^{(n)} \neq EOL$ **do**
      if $s_i^{(n)} \in dict$ *and* $\alpha >= r$ **then**
        $tgt_i^{(n)} \leftarrow dict[s_i^{(n)}]$;
        $\alpha \leftarrow 0$;
      else
        $tgt_i^{(n)} \leftarrow s_i^{(n)}$;
        $\alpha \leftarrow \alpha + 1$;
      end
      $i \leftarrow i + 1$;
    end
    $t \leftarrow t \cup tgt^{(n)}$;
  end
end
$\Phi \leftarrow t \cup S$;

C. CHALLENGES IN CODE MIX WORD EMBEDDING

CM sentences pose a number of challenges in terms of appropriately representing them in embedding space. ***(1) Spelling errors:*** CM text, being informal, is prone to mistakes such as misspellings. These deviations cannot be ignored as typographical errors since they convey sentiments. For example, *good* conveys positive sentiment, as does *gd, gooood*. As a result, while doing tasks like sentiment analysis, we must take these variations into consideration. ***(2) Out-of-vocabulary:*** In general, we train the model with a finite set of words, but in real world, when dealing with causal messaging, individuals utilize a range of words that are not included in the lexicon. This type of out-of-vocabulary usage is common in real-world applications. Word2vec and glove embedding approaches, however, are incapable of handling such out-of-bag vocabulary. ***(3) Alignment Process:*** Another difficulty with CM data is that we have different monolingual representations for the same word that should be represented in

the close proximity. For instance, since *good* represents the same context as *bala* or *bhalo*, they should be located near to one another in the embedding space. When CM sentences are embedded similarly as a single monolingual text, the cross-lingual relationships between words are disregarded. Therefore, we must handle CM texts differently than generic texts. To address the challenges depicted in ***(1)*** and ***(2)***, we implement the Fasttext model [10]. In section IV-B, we discussed how the Fasttext model correctly captures subword level information, thereby accounting for spelling errors or word variations while being applicable to out of bag vocabulary.

To address the challenge posed by ***(3)*** for encoding the cross-lingual representation, we proposed a simple yet effective data augmentation technique (Sec. IV-D), in which we extract selected words from the text and replace them with their monolingual equivalents using a dictionary,[3] followed by the Fasttext word embedding. This transliterated Bangla-English dictionary allows us to substitute words with their monolingual counterpart. However, online translators are also available to convert one word to another, but transliterated Bangla words have different phonetics than Bangla. Therefore, we have to collect this transliterated dictionary.

D. PROPOSED DATA AUGMENTATION

We extend the dictionary based code switching data augmentation in [60] and utilize it for current monolingual word embeddings in order to develop cross-lingual understanding. The proposed data augmentation process (See Figure 5(a)) is as follows: Firstly, to begin, apart from [60], we define the sampling rate parameter (r), which determines how frequently we select words to update from the source language to the target language, instead of randomly sampling words from a sentence, which gives us more flexibility to enhance the cross-lingual alignment. If the selected word is in Bangla (source language), we transform it to its English (target language) equivalent and vice versa. We employ several sample rates to further improve the quality of data augmentation. Whereas [60] defined a single sampling rate (random selection), we employ a hierarchical sampling rate to improve the cross-lingual alignment since from our study, we found that considering only a single sampling rate or random word selection [60] results in a poor cross-lingual adaptation for embedding method (See Fig 7(b), Tab 8) since it heavily overlooks a certain group of words by not converting them to their monolingual counterpart and only consider words that satisfy the specific sampling rate parameter. This approach, therefore, decreases the quality of the cross-lingual alignment. Moreover, this problem can be expected to be worse for low resource languages with limited corpus size since we are not utilizing a sentence to its full potential. Furthermore, note that at first glance it seems lucrative to implement only the sampling rate to 0 since this will convert every single word into its monolingual counterpart without loss of information (when we consider $r > 0$ we ignore some group of words thus losing some information). However, converting every context word into their monolingual counterpart (English or Bengali) results in a separate clustering for the respective languages. To overcome these aforementioned limitations, we propose to implement hierarchical sampling rates, where we consider the majority of the words within a sentence and convert them to their monolingual counterpart through multiple sampling rates and can effectively handle the information loss while also improving the cross-lingual alignment for current word embedding models which is supported by the experimental results from Fig 7(c), Tab 8. Moreover, [60] employ the data augmentation strategy with randomly selected sentences that incur additional information loss for low resource languages instead of picking random sentences to perform augmentation, we propose to employ this data augmentation strategy for every sentence. Finally, unlike the work in [60], we also include the original text in our augmented corpus, which further increase the diversity of the training sample. With the augmented training data at hand, we train the word embedding model (e.g., Fasttext [10]) as shown in Figure 5(b) and feed the word representation (i.e., the output of the word embedding model) to a classifier for sentiment prediction (See Figure 5(c)). Intuitively, training with the augmented dataset can make the model automatically align the replaced word in the target language and the original word in a source language into a similar vector space according to their similar context information. As a result, our proposed alignment with Fasttext word embedding successfully resides cross-lingual information (i.e. the distance between adjacent multilingual words is small), as seen in Figure 6(b), whereas Fasttext model without data augmentation fails to do so, as illustrated in Figure 6(a).

The pseudo-code for the proposed data augmentation process is shown in Algorithm 1. The algorithm, in Lines 5-7, determines whether the i-th word in the n-th sentence is to be selected for replacement depending on the sampling rate. If the word gets selected, it is replaced by its corresponding word in the target language and stored in a temporary variable tgt_i^n. Otherwise, lines 9-10 are executed, and the word is stored in tgt_i^n as it is. The algorithm also increments the variable α which is used as a counter to count the gaps between each selected word for replacement. The code then iterates to the next word in line 12. Thus, for each sampling rate, the code goes through each word within n-th sentence, and stores the augmented sentences at $\mathbf{t}$ (line 14). This operation is performed for all sentences in the data set and finally in line 17 the augmented sentences along with original sentences are stored in Φ, which is utilized to train the word embedding model.

V. EXPERIMENTAL DETAILS

A. BASELINES

To analyze the performance boost associated with our proposed data augmentation method, we compared it to the

[3]https://github.com/diptamath/Language-Identification-of-Bengali-English-Code-Mixed-data-using-LSTM

most widely used models for sentiment classification. These models fall mainly into the machine learning (ML) and deep learning (DL) categories. We chose Logistic Regression, Decision Tree, Support Vector Machine, and Extreme Gradient Boosting as machine learning models and used Word2vec, FastText, and cm-FastText embeddings as input. For the remainder of the paper, we will refer to the FastText model trained with our proposed augmented data as cm-FastText. In the case of DL models, we chose 1DConv-LSTM, BERT-Multilingual, Distill Bert, and Base Bert, where the embeddings are the same as before. These baselines are as follows:

1) LOGISTIC REGRESSION (LR) [12]

We apply the Logistic Regression model with L2 regularization.

2) SUPPORT VECTOR MACHINE (SVM)[12]

We use the SVM model with L2 regularization. The purpose of the SVM classification algorithm is to define an optimal hyperplane in N-dimensional space to separate the data points from each other.

3) DECISION TREE (DT)[14]

A decision tree splits the total training set into several subsets as nodes, and each node attempts to predict the label. After sequentially choosing alternative decisions, each node is recursively split again, and finally, the classifier defines rules based on criteria to predict the final result. We used decision trees with a maximum depth of 800 and a minimum of 5 sample splits.

4) EXTREME GRADIENT BOOSTING (XGBoost)[14]

XGBoost is an implementation of gradient boosting with several additional features focused on performance and speed[15]. We used 500 estimator trees in our experiment.

5) BASE BERT (BASE-BERT)[16]

Base BERT, introduced in [16], is a transformer [73] based model pre-trained only on English data in a self-supervised fashion without human labelling.

6) BERT-MULTILINGUAL(m-BERT)[18]

It is a pre-trained model on unlabelled text from multiple languages and can be fine-tuned further by adding a classification layer. BERT has been used for many text classification tasks [74], [75], [76].

7) DISTILL BERT(DISTIL-BERT)[17]

It learns a simplified version of Base BERT that retains 97% performance but uses half the parameters. Distil BERT employs distillation to approximate the Base BERT. Once trained, a large neural network's output distribution can be approximated by a smaller network.

8) 1DConv-LSTM

The model consists of an Embedding layer, Dropout with probability 0.3, single conv1D layer with 5×1 filter size with ReLU activation, 2×1 1DMax-pooling, one LSTM layer and a classifier with the softmax activation.

B. IMPLEMENTATION DETAILS

Instead of a single sample rate, we employ several sampling rates, specifically 1, 2, 3, to train the word embedding models in the data augmentation. We use 1DConv-LSTM with the Adam optimizer for 25 epoaches with a learning rate of 0.001, batch size of 64, Relu activation function, and cross entropy loss. The bert models are implemented using Hugging Face.[4] We implement the baselines (LR, SVM, and DT) in *sklearn*.[5] We report the average test performance over 10 runs to evaluate the performance of the algorithms fairly. We performed both training and testing ten times.

C. EVALUTATION METRICS

Precision: It is the ratio of correctly predicted positive observations to the total predicted positive observations. It is defined as follows:

$$\text{Precesion} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

Recall: Recall is defined as:

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

F1 score: The F1 score is interpreted as a harmonic mean of the precision and recall. As a result, it considers both false positives and false negatives. It is defined as:

$$\text{F1 score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

It is worth noting that the weighted-averaged F1 score is calculated by taking the mean of all per-class F1 scores while considering each class's contribution. Therefore, weighted F1 score is more expressive compared to other evaluation metrics (e.g., accuracy, precision, and recall) in case of unevenly distributed dataset since it considers each class's contribution.

D. OVERALL RESULTS

We show the results of sentiment classification in terms of precision (P), recall (R), F1-score (F), and weighted F1 score (WF) for the baseline methods in Table 5. Since, our dataset is highly imbalanced, with the majority of sentiment being either positive or negative, WF is a better evaluation metric in this case to judge the performance of the models.

As shown in Table 5, our proposed data augmentation method aids all models under study in improving their performance in terms of weighted F1 score by capturing cross-lingual relationships between similar words in different

[4]https://huggingface.co/docs/transformers/
[5]https://scikit-learn.org/

TABLE 5. Results from supervised learning experiments using several algorithms on our dataset. The **bold** font denote the best performance for each method across all word embeddings whereas the red rectangle denote the best overall performance.

Method	Embedding	Neg.			Mix. neg.			Neutral			Mix. pos.			Pos.			WF
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	
LR	Word2vec	0.81	0.89	0.85	0.58	0.23	0.33	0.33	0.15	0.21	0.33	0.14	0.19	0.68	0.71	0.69	0.74
	cm-Word2vec	0.84	0.88	0.86	0.52	0.44	0.48	0.45	0.36	0.41	0.43	0.25	0.32	0.71	0.71	0.71	0.76
	Fasttext	0.81	0.89	0.85	0.52	0.27	0.35	0.36	0.17	0.23	0.37	0.17	0.23	0.70	0.72	0.71	0.74
	cm-Fasttext	0.85	0.87	0.86	0.56	0.45	0.50	0.47	0.41	0.44	0.40	0.32	0.36	0.72	0.74	0.73	**0.77**
SVM	Word2vec	0.78	0.94	0.85	1.00	0.03	0.07	0.71	0.03	0.05	0.00	0.00	0.00	0.69	0.67	0.68	0.71
	cm-Word2vec	0.79	0.95	0.86	0.96	0.14	0.24	0.81	0.09	0.16	0.87	0.07	0.13	0.74	0.71	0.72	0.74
	Fasttext	0.77	0.95	0.85	1.00	0.01	0.02	0.67	0.01	0.02	0.00	0.00	0.00	0.71	0.66	0.68	0.71
	cm-Fasttext	0.80	0.96	0.87	1.00	0.13	0.23	0.85	0.06	0.11	0.67	0.04	0.08	0.76	0.70	0.73	**0.75**
DT	Word2vec	0.88	0.83	0.85	0.57	0.73	0.64	0.49	0.58	0.53	0.37	0.43	0.40	0.71	0.71	0.71	0.78
	cm-Word2vec	0.86	0.86	0.86	0.49	0.64	0.56	0.54	0.51	0.52	0.39	0.31	0.35	0.74	0.72	0.73	0.78
	Fasttext	0.87	0.85	0.86	0.59	0.72	0.64	0.48	0.59	0.53	0.35	0.40	0.37	0.72	0.70	0.71	0.78
	cm-Fasttext	0.87	0.84	0.86	0.55	0.75	0.65	0.47	0.58	0.52	0.44	0.44	0.44	0.73	0.71	0.72	**0.79**
XGBoost	Word2vec	0.87	0.94	0.90	0.95	0.66	0.78	0.85	0.51	0.64	0.78	0.39	0.52	0.81	0.78	0.79	0.85
	cm-Word2vec	0.87	0.95	0.91	0.95	0.71	0.81	0.84	0.56	0.67	0.78	0.39	0.52	0.82	0.78	0.80	0.86
	Fasttext	0.87	0.96	0.91	0.99	0.71	0.83	0.85	0.49	0.62	0.78	0.46	0.58	0.83	0.78	0.81	0.86
	cm-Fasttext	0.87	0.96	0.92	0.98	0.71	0.82	0.87	0.50	0.63	0.81	0.46	0.59	0.85	0.79	0.82	**0.87**
m-BERT		0.83	0.96	0.89	0.52	0.10	0.16	0.31	0.25	0.28	0.00	0.00	0.00	0.82	0.75	0.78	**0.78**
cm-m-BERT		0.84	0.94	0.89	0.35	0.21	0.27	0.37	0.14	0.21	0.36	0.28	0.32	0.81	0.74	0.77	**0.78**
distil-BERT		0.80	0.94	0.87	0.00	0.00	0.00	0.12	0.13	0.13	0.00	0.00	0.00	0.80	0.68	0.74	0.73
cm-distil-BERT		0.81	0.95	0.87	0.00	0.00	0.00	0.29	0.09	0.14	0.00	0.00	0.00	0.73	0.74	0.73	**0.74**
base-BERT		0.87	0.93	0.90	0.46	0.33	0.38	0.40	0.30	0.34	0.49	0.20	0.28	0.80	0.81	0.80	0.81
cm-base-BERT		0.88	0.93	0.90	0.66	0.53	0.59	0.63	0.38	0.47	0.46	0.35	0.40	0.79	0.78	0.79	**0.83**
1DConv-LSTM	Word2vec	0.90	0.87	0.88	0.65	0.71	0.68	0.72	0.57	0.64	0.50	0.40	0.45	0.71	0.81	0.76	0.82
	cm-Word2vec	0.87	0.92	0.89	0.68	0.68	0.68	0.82	0.48	0.60	0.52	0.42	0.47	0.78	0.75	0.76	0.83
	Fasttext	0.89	0.89	0.89	0.68	0.69	0.69	0.74	0.51	0.61	0.50	0.41	0.45	0.75	0.80	0.77	0.83
	cm-Fasttext	0.88	0.92	0.90	0.86	0.64	0.74	0.63	0.55	0.59	0.55	0.40	0.47	0.79	0.77	0.78	**0.84**

TABLE 6. Results for zero shot cross lingual adaptation with 1DConv-LSTM on CM sentiment analysis task. The red rectangle highlights the word embedding with the best performance.

Embedding	Neg.			Mix. neg.			Neutral			Mix. pos.			Pos.			WF
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	
Word2vec	0.660	0.690	0.664	0.064	0.070	0.062	0.056	0.044	0.048	0.040	0.098	0.054	0.462	0.256	0.262	0.492
cm-Word2vec	0.678	0.812	0.738	0.054	0.026	0.034	0.074	0.036	0.044	0.046	0.034	0.04	0.444	0.298	0.344	0.556
Fasttext	0.662	0.770	0.706	0.07	0.032	0.040	0.058	0.016	0.022	0.05	0.101	0.062	0.406	0.224	0.272	0.522
cm-Fasttext	0.682	0.854	0.758	0.060	0.042	0.044	0.090	0.046	0.060	0.058	0.046	0.042	0.538	0.250	0.328	0.568

languages. The following are our additional observations from Table 5. *Firstly*, we found that the precision, recall and F1-scores for *negative* and *positive* classes are much higher than that of the *Mixed negative* and *Mixed positive* classes using both ML and DL-based approaches. To be specific, we demonstrated in Figure 2 that 23.4% and 63.9% of total 18,074 sentences, belong to the *Positive* and *Negative* sentiment classes respectively, whereas the remaining sentiment classes account for 4.8%, 5.0% and 2.8%. Additionally, *Mixed negative* and *Mixed positive* contain implicit cues that make discrimination challenging in some cases, even for humans. Positive and negative classes, on the other hand, convey more specific cues that result in higher performance.

Secondly, among the ML models, XGBoost acquires the highest score for Fasttext embedding trained with our proposed data augmentation method (cm-FastText) compared to other word embeddings. In our experiments, we observe that tree-based approaches such as DT perform relatively well in the majority of performance metrics. This conclusion is also confirmed by XGBoost's performance, which outperforms the rest of the algorithms in Table 5. We also found that, in the case of Word2vec and Fasttext word embeddings, LR and SVM models perform poorly on classes with a limited sample size. However, cm-FastText shines for LR and SVM models in that case, since training on our augmented data can increase the distribution of instances by utilizing different sampling

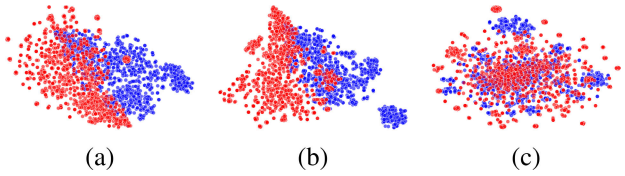

FIGURE 7. t-SNE[79] of words in embedding space shown as a two-dimensional vector. Blue color indicates English words, whereas red color denotes the Bangla-counterpart. (a) word vector representation when we convert every word into it's monolingual counterpart, (b) data augmentation with random word selection as proposed in [60], (c) our proposed data augmentation with sampling rate, r = 1, 2 and 3.

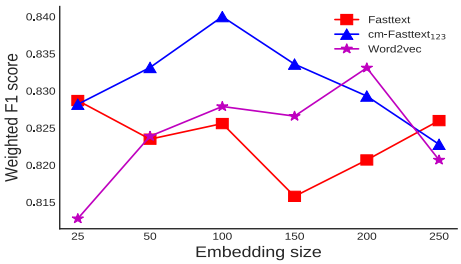

FIGURE 8. Hyper-parameter validation on different word embedding methods and embedding size.

rates. *Thirdly*, in DL-based approaches, 1DConv-LSTM and base-Bert perform relatively well across all five sentiment classes. Among the BERT models, although, m-BERT is trained on 106 distinct languages [18], it lacks samples of transliterated Bengali text. In this work, we implemented these BERT-based models while only fine-tuning the parameters of the pre-trained BERT models with our proposed code-mixed dataset. As a result, with smaller training parameters (110 millions), the BERT Base model [77], [78] outperforms the m-BERT that has a greater number of learnable parameters (178 million), implying that m-BERT requires more training data to train. This observation is reinforced by the 1DConv-LSTM model, where it outperforms all Bert-based architectures since it is a smaller architecture with the fewest number of learnable parameters as compared to Bert models. However, Distill-BERT [17] is the smallest architecture among the Bert models (40% smaller, 60% faster), but it still underperforms Bert-base architecture since it is a distill approximation of Bert architecture, where the performance is upper-bound by the actual Bert-base model. *Finally*, when we compare each algorithm with respect to word embeddings, we notice that Fasttext combined with our proposed data augmentation method (cm-FastText) outperforms the baseline models for the majority of classes. The reason for this is that word2vec embedding learns to capture a word's semantic information based on its context. However, it does not take into account word variations and out-of-bag vocabulary words. Fasttext overcomes this by learning the subword level word embeddings and combining them all to anticipate the final word embeddings. As a result, it is capable of capturing word variants and out-of-bag vocabulary in the text. The overall outcome is improved further by our data augmentation process, which captures both word variations and cross-lingual information.

1) ZERO SHOT CROSS LINGUAL ADAPTATION ON CM SENTIMENT CLASSIFICATION

Similar to [11] and [60], in the zero shot cross lingual adaptation for CM sentiment classification setup, the baseline model (1DConv-LSTM) is trained on the English language for sentiment analysis and performs a zero-shot cross-lingual transfer task by directly predicting sentiment for the code mixed Bangla. This task's performance is highly dependent on the cross-linguistic quality of the word embeddings. Given the huge number of monolingual (English) sentiment text product reviews available online, we extracted these reviews with annotation from the Google Playstore to serve as our training sample for this experiment. We choose 1DConvLSTM only as the baseline model since our purpose is to demonstrate the cross lingual quality of the existing monolingual word embeddings when combine with proposed data augmentation. The experimental results for zero-shot classification in Table 6 reveal that Fasttext and Word2vec with our data augmentation strategy substantially outperform the same word embedding models without data-augmentation. The reason for this is that our data augmentation technique significantly improves the cross-lingual quality of monolingual word embeddings.

E. ABLATION STUDY

1) HYPER-PARAMETER TUNING

We tune hyperparameters on a separate validation set. Within the fit() function of tensorflow 2.0, we set the validation split to be 0.2 of the whole training set. We validate the model for embedding size k within the range $\{25, 50, 100, 150, 200\}$ and the number of sampling rates within the range $\{1, 2, 3, [1, 2, 3]\}$, where $[1, 2, 3]$ represents multiple sampling rates such as 1, 2 and 3. We use $k = 100$ for our final model based on the validation experiments. It demonstrates that raising the embedding size from 25 to 100 improves performance for cm-Fasttext, but further increasing the embedding size makes no noticeable improvement. We provide experimental result with different word embedding method and embedding size in Figure 8.

2) FEATURE VISUALIZATION

In order to see whether training word embedding model with our proposed data augmentation method can align the representation between the source language and the target language we perform the t-SNE [79] visualization of the word vectors. From Fig 7(a), when we train the monlingual Fasttext model while converting every word to it's monolingual counterpart, we see that there is no overlap between words of different languages (English and Bangla), which show that there is a lack of cross lingual adaptation since it treat similar word pairs form different languages differently. In

TABLE 7. Comparison between different sampling rates and embedding size. Fasttext performed best when we used a combined 1, 2, 3 sampling rate in the augmented corpus and an embedding size of 100.

Augmentation	Embedding size					
	25	50	100	150	200	250
cm-Fasttext$_1$	0.825	0.826	0.821	0.825	0.833	0.834
cm-Fasttext$_2$	0.825	0.805	0.832	0.828	0.820	0.830
cm-Fasttext$_3$	0.818	0.830	0.833	0.823	0.818	0.818
cm-Fasttext$_{123}$	0.828	0.833	**0.841**	0.834	0.829	0.823

TABLE 8. Model efficacy analysis with the Word2Vec embedding. Word2vec performs the best when we used a combined 1, 2, and 3 sampling rate, here the subindex indicates the sampling rate, and *W2V* represents the Word2vec model trained with the original training set without data augmentation.

Augmentation	precision	recall	F1-score
W2V	0.822	0.813	0.813
W2V$_3$	0.824	0.825	0.824
W2V$_{23}$	0.829	0.825	0.827
W2V$_{123}$	**0.832**	**0.835**	**0.833**
W2V$_{1234}$	0.829	0.833	0.828

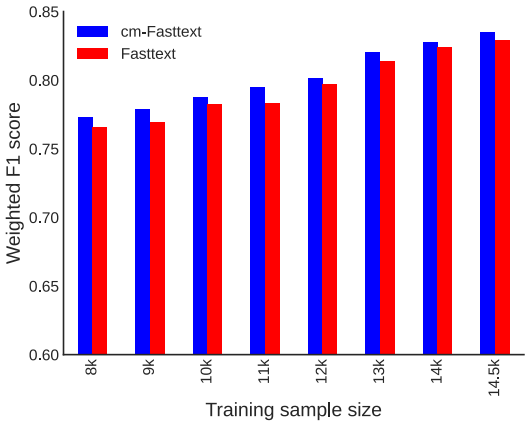

FIGURE 9. F1 score when adding training data for different dataset sizes on 1DConv-LSTM model. In each case, our proposed augmentation shows better performance than the traditional word-embedding.

constrast, in Fig 7(b), when we implement our proposed data augmentation strategy proposed as [60], we observe some overlapping regions between different languages that indicate cross lingual adaptation between similar word pairs of two different languages. However,in Fig 7(b) we can still clearly distinguish two different clusters formation. This cross lingual alignment is further improved with our proposed hierarchical sampling rate in fig 7(c), where two different clusters that represent two different languages merge together with a single cluster formation with highest overlapping regions, which further demonstrates that our hierarchical sampling rate can effectively and successfully aligns representations of different languages closer.

3) EFFECT OF SAMPLING RATES

To demonstrate the influence of sampling rate, we conducted the experimental study shown in Table 7. We report the weighted F1 score as the performance metric. We experiment with sampling rates, $r = \{1, 2, 3, [1, 2, 3]\}$, where the size of the embedding vector is {25, 50, 100, 150, 200, 250}. Rather of utilizing a single sampling rate, we discover that employing several sampling rates improves the quality of the augmented sentences, hence improving the overall performance.

4) MODEL EFFICACY ANALYSIS

To assess the efficacy of our proposed data augmentation we further conduct an experiment with two different datasets but of the same size: ***(1)*** the original training set, and ***(2)*** the augmented data set, while maintaining the same size by using only a single sampling rate, and excluding the original sentences from the training set. We observe that when Word2Vec is trained with our proposed data augmentation ($W2V_3$, where the subscript represents sampling rate) the inference performance of the sentiment classifier is marginally increased as compared to the baseline $W2V$ that is trained without any data augmentation (See Table 8). The reason is with our proposed data augmentation strategy at its minimalistic form is also able to improve the cross-lingual adaption (See Fig 7(b)) compare to $W2V$, thus improving the overall performance for $W2V_3$. However, as we change the corpus size by utilizing multiple sampling rates ($W2V_{23}$ and $W2V_{123}$) to train the word2vec model, we observe gradual improvements in the performance. The best performance is obtained with hierarchical sampling rates 1,2 and 3 ($W2V_{123}$) as shown in Table 8. The reason is that our proposed hierarchical sampling rate enhanced the cross-lingual adaptation of word vectors representation, where similar words of two different languages are in a close cluster with maximum overlapping regions (See Fig 7(c)). As a result, the performance of the final classifier is improved by this improved cross-lingual alignment. However, further changing the training size with a sampling rate of 4 to train the Word2Vec model represented as $W2V_{1234}$ does not result in an improvement in performance due to redundant repetition in word switching.

5) VARYING AMOUNTS OF TRAINING DATA

We study the effectiveness of our proposed method by varying the size of the training data. We simulate the setup by sub-sampling the training dataset where we started with 8000 training samples and successively add 500 samples for each sub-sample training dataset. The test dataset remains the same as our previous setup. Then we train the model on the sub-sampled training dataset with both the Fasttext and ours' cm-Fasttext embedding and evaluated the weighted F1 score of the test set in Fig 9. This weighted F1 score steadily increases as we gradually increase training data in both normal Fasttext and cm-Fasttext. This is likely because adding more samples increases diversity which decreases the generalization error. On normal Fasttext setup with 1DConv-LSTM, we measure 76.5% for 8k, 79.6% for 12k and 82.9%

for 14.5k samples. However, cm-Fasttext with 1DConv-LSTM constantly outperforms the baseline in all sub-sample training dataset. Fasttext with proposed augmentation performed better with 77.3% for 8k sample, 80.2% for 12k, and 83.8% for 14.5k.

VI. CONCLUSION

This work presents a data augmentation approach for code-mixed sentences that helps monolingual word embeddings in aligning representations across multiple languages that share a similar context. Experimental evidence suggests that understanding this cross-lingual relationship can aid in improving the overall performance of existing sentiment classifiers. Furthermore, in current literature, there is a scarcity of resources for studying the code mixing phenomenon. The majority of earlier studies were done on monolingual corpora. To address this, we developed a gold-standard Bangla-English sentiment analysis corpus. This resource can also serve as a starting point for future researchers interested in investigating the code mixing phenomenon.

To conclude, we believe that incorporating cross-lingual embedding into sentiment analysis for code-mixed data is a feasible approach. Due to the prevalence of code mixing in multilingual societies such as Hindi, Arabic, and Bangla, the concerns raised in this paper are applicable to a broad variety of languages and tasks. As a consequence, the proposed approach may be applied to code-mixed text processing in a variety of languages and may make a substantial contribution to addressing the data-acquisition bottleneck in code-mixed data.

ACKNOWLEDGMENT

The authors express their deep gratitude to Apurba Roy, Junayed Ahmed, Mohammad Shams, Saniah Kayenat Chowdhury, Shudipto Pramanic, Swapno Nila Shreya, and Utsha Kumar Roy for their enormous support in data annotation.

REFERENCES

[1] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2010, pp. 939–948.

[2] X. Fang and J. Zhan, "Sentiment analysis using product review data," *J. Big Data*, vol. 2, no. 1, pp. 1–14, Dec. 2015.

[3] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 61, no. 12, pp. 2544–2558, Dec. 2010.

[4] A. Jamatia, S. D. Swamy, B. Gambäck, A. Das, and S. Debbarma, "Deep learning based sentiment analysis in a code-mixed English-Hindi and English-Bengali social media corpus," *Int. J. Artif. Intell. Tools*, vol. 29, no. 5, Aug. 2020, Art. no. 2050014.

[5] S. Mandal, S. K. Mahata, and D. Das, "Preparing Bengali-English code-mixed corpus for sentiment analysis of Indian languages," 2018, *arXiv:1803.04000*.

[6] A. Elia, S. Pelosi, A. Maisto, and R. Guarasci, "Towards a lexicon-grammar based framework for NLP: An opinion mining application," in *Proc. Int. Conf. Recent Adv. Natural Lang. Process.*, 2015, pp. 160–167.

[7] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. Linguistics*, vol. 37, no. 2, pp. 267–307, Jun. 2011.

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Adv. neural Inf. Process. Syst.*, vol. 26, 2013.

[9] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2017.

[11] S. Yadav and T. Chakraborty, "Zero-shot sentiment analysis for code-mixed data," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 18, pp. 15941–15942.

[12] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4, no. 4. New York, NY, USA: Springer, 2006.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1997.

[14] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 112. New York, NY, USA: Springer, 2013.

[15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[17] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.

[18] T. Pires, E. Schlinger, and D. Garrette, "How multilingual is multilingual BERT?" 2019, *arXiv:1906.01502*.

[19] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2004, pp. 168–177.

[20] S. Akter and M. T. Aziz, "Sentiment analysis on Facebook group using lexicon based approach," in *Proc. 3rd Int. Conf. Electr. Eng. Inf. Commun. Technol. (ICEEICT)*, Sep. 2016, pp. 1–4.

[21] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. J. Passonneau, "Sentiment analysis of Twitter data," in *Proc. Workshop Lang. Social media (LSM)*, 2011, pp. 30–38.

[22] T. Chalothom and J. Ellman, "Simple approaches of sentiment analysis via ensemble learning," in *Information Science and Applications*. Berlin Germany: Springer, 2015, pp. 631–639.

[23] M. Koehler, S. Greenhalgh, and A. Zellner, "Potential applications of sentiment analysis in educational research and practice—Is site the friendliest conference?" in *Proc. Soc. Inf. Technol. Teacher Educ. Int. Conf.* Asheville, NC, USA: Association for the Advancement of Computing in Education (AACE), 2015, pp. 1348–1354.

[24] B. Song, C. Lee, B. Yoon, and Y. Park, "Diagnosing service quality using customer reviews: An index approach based on sentiment and gap analyses," *Service Bus.*, vol. 10, no. 4, pp. 775–798, Dec. 2016.

[25] S. Zhong-Feng and W. Jing, "RCNN-BGRU-HN network model for aspect-based sentiment analysis," *Comput. Sci.*, vol. 46, no. 9, pp. 223–228, 2019.

[26] H. Hongye, Z. Jin, and Z. Zuping, "Text sentiment analysis combined with part of speech features and convolutional neural network," *Comput. Eng.*, vol. 44, no. 11, pp. 215–220, 2018.

[27] S. Chowdhury and W. Chowdhury, "Performing sentiment analysis in Bangla microblog posts," in *Proc. Int. Conf. Informat., Electron. Vis. (ICIEV)*, May 2014, pp. 1–6.

[28] S. Islam, A. Islam, A. Hossain, and J. J. Dey, "Supervised approach of sentimentality extraction from Bengali Facebook status," in *Proc. 19th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2016, pp. 383–387.

[29] T. Ghosal, S. K. Das, and S. Bhattacharjee, "Sentiment analysis on (Bengali horoscope) corpus," in *Proc. Annu. IEEE India Conf. (INDICON)*, Dec. 2015, pp. 1–6.

[30] S. A. Mahtab, N. Islam, and M. M. Rahaman, "Sentiment analysis on Bangladesh cricket with support vector machine," in *Proc. Int. Conf. Bangla Speech Lang. Process. (ICBSLP)*, Sep. 2018, pp. 1–4.

[31] K. Sarkar and S. Chakraborty, "A sentiment analysis system for Indian language tweets," in *Proc. Int. Conf. Mining Intell. Knowl. Explor.* Springer, 2015, pp. 694–702.

[32] K. Sarkar and M. Bhowmick, "Sentiment polarity detection in Bengali tweets using multinomial Nïive Bayes and support vector machines," in *Proc. IEEE Calcutta Conf. (CALCON)*, Dec. 2017, pp. 31–36.

[33] A. Hassan, M. R. Amin, A. K. A. Azad, and N. Mohammed, "Sentiment analysis on Bangla and romanized Bangla text using deep recurrent models," in *Proc. Int. Workshop Comput. Intell. (IWCI)*, Dec. 2016, pp. 51–56.

[34] H. Ali, Md. F. Hossain, S. B. Shuvo, and A. Al Marouf, "BanglaSenti: A dataset of Bangla words for sentiment analysis," in *Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2020, pp. 1–4.

[35] K. Singh, I. Sen, and P. Kumaraguru, "A Twitter corpus for Hindi-English code mixed POS tagging," in *Proc. 6th Int. Workshop Natural Lang. Process. Social Media*. Melbourne, VIC, Australia: Association for Computational Linguistics, Jul. 2018, pp. 12–17.

[36] S. Si, A. Datta, S. Banerjee, and S. K. Naskar, "Aggression detection on multilingual social media text," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–5.

[37] T. Y. S. S. Santosh and K. V. S. Aravind, "Hate speech detection in Hindi-English code-mixed social media text," in *Proc. ACM India Joint Int. Conf. Data Sci. Manage. Data*, Jan. 2019, pp. 310–313.

[38] R. Bhargava, Y. Sharma, and S. Sharma, "Sentiment analysis for mixed script indic sentences," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2016, pp. 524–529.

[39] S. Sazzed and S. Jayarathna, "A sentiment classification in Bengali and machine translated English corpus," in *Proc. IEEE 20th Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Jul. 2019, pp. 107–114.

[40] E. Pustulka-Hunt, T. Hanne, E. Blumer, and M. Frieder, "Multilingual sentiment analysis for a Swiss gig," in *Proc. 6th Int. Symp. Comput. Bus. Intell. (ISCBI)*, Aug. 2018, pp. 94–98.

[41] M. J. Fuadvy and R. Ibrahim, "Multilingual sentiment analysis on social media disaster data," in *Proc. Int. Conf. Electr., Electron. Inf. Eng. (ICEEIE)*, vol. 6, Oct. 2019, pp. 269–272.

[42] B. Haidar, M. Chamoun, and A. Serhrouchni, "A multilingual system for cyberbullying detection: Arabic content detection using machine learning," *Adv. Sci., Technol. Eng. Syst. J.*, vol. 2, no. 6, pp. 275–284, Dec. 2017.

[43] A. Suciati and I. Budi, "Aspect-based opinion mining for code-mixed restaurant reviews in Indonesia," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Nov. 2019, pp. 59–64.

[44] V. Singh, D. Vijay, S. S. Akhtar, and M. Shrivastava, "Named entity recognition for Hindi-English code-mixed social media text," in *Proc. 7th Named Entities Workshop*. Melbourne, VIC, Australia: Association for Computational Linguistics, Jul. 2018, pp. 27–35.

[45] M. Singh, V. Goyal, and S. Raj, "Sentiment analysis of English-Punjabi code mixed social media content for agriculture domain," in *Proc. 4th Int. Conf. Inf. Syst. Comput. Netw. (ISCON)*, Nov. 2019, pp. 352–357.

[46] A. Pratapa, G. Bhat, M. Choudhury, S. Sitaram, S. Dandapat, and K. Bali, "Language modeling for code-mixing: The role of linguistic theory based synthetic data," in *Proc. 56th Annu. Meeting Assoc. for Comput. Linguistics*, vol. 1. Melbourne, VIC, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1543–1553. [Online]. Available: https://aclanthology.org/P18-1143

[47] Y. Vyas, S. Gella, J. Sharma, K. Bali, and M. Choudhury, "POS tagging of English-Hindi code-mixed social media content," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 974–979.

[48] M. Faruqui and C. Dyer, "Improving vector space word representations using multilingual correlation," in *Proc. 14th Conf. Eur. Chapter Assoc. Comput. Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014, pp. 462–471. [Online]. Available: https://aclanthology.org/E14-1049

[49] K. M. Hermann and P. Blunsom, "Multilingual models for compositional distributed semantics," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*. Baltimore, MD, USA: Association for Computational Linguistics, Jun. 2014, pp. 58–68. [Online]. Available: https://aclanthology.org/P14-1006

[50] T. Luong, H. Pham, and C. D. Manning, "Bilingual word representations with monolingual quality in mind," in *Proc. 1st Workshop Vector Space Modeling Natural Lang. Process.* Denver, CO, USA: Association for Computational Linguistics, Jun. 2015, pp. 151–159. [Online]. Available: https://aclanthology.org/W15-1521

[51] X. Zhou, X. Wan, and J. Xiao, "Cross-lingual sentiment classification with bilingual document representation learning," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2016, pp. 1403–1412.

[52] X. Chen, Y. Sun, B. Athiwaratkun, C. Cardie, and K. Weinberger, "Adversarial deep averaging networks for cross-lingual sentiment classification," *Trans. Assoc. Comput. Linguistics*, vol. 6, pp. 557–570, Dec. 2018.

[53] S. Ruder, I. Vulić, and A. Søgaard, "A survey of cross-lingual word embedding models," *J. Artif. Intell. Res.*, vol. 65, pp. 569–631, Aug. 2019.

[54] A. Klementiev, I. Titov, and B. Bhattarai, "Inducing crosslingual distributed representations of words," in *Proc. COLING*, 2012, pp. 1459–1474.

[55] T. Kočiský, K. M. Hermann, and P. Blunsom, "Learning bilingual word representations by marginalizing alignments," 2014, *arXiv:1405.0947*.

[56] J. Guo, W. Che, D. Yarowsky, H. Wang, and T. Liu, "A representation learning framework for multi-source transfer parsing," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 1–7.

[57] C. Xing, D. Wang, C. Liu, and Y. Lin, "Normalized word embedding and orthogonal transform for bilingual word translation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 1006–1011.

[58] G. Lample, A. Conneau, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=H196sainb

[59] W. Chen, J. Chen, Y. Su, X. Wang, D. Yu, X. Yan, and W. Y. Wang, "XL-NBT: A cross-lingual neural belief tracking framework," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Brussels, Belgium: Association for Computational Linguistics, Oct./Nov. 2018, pp. 414–424. [Online]. Available: https://aclanthology.org/D18-1038

[60] L. Qin, M. Ni, Y. Zhang, and W. Che, "CoSDA-ML: Multi-lingual code-switching data augmentation for zero-shot cross-lingual NLP," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, Jul. 2020, pp. 3853–3860, doi: 10.24963/ijcai.2020/533.

[61] H. Huang, Y. Liang, N. Duan, M. Gong, L. Shou, D. Jiang, and M. Zhou, "Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 2485–2494.

[62] S. Wu and M. Dredze, "Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 833–844. [Online]. Available: https://aclanthology.org/D19-1077

[63] A. Conneau and G. Lample, "Cross-lingual language model pretraining," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–10.

[64] M. Artetxe and H. Schwenk, "Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 597–610, Nov. 2019.

[65] M. A. A. Mumin, M. H. Seddiqui, M. Z. Iqbal, and M. J. Islam. (2018). *SUPara0.8M: A Balanced English-Bangla Parallel Corpus*. [Online]. Available: https://ieee-dataport.org/documents/supara08m-balanced-english-bangla-parallel-corpus

[66] T. Hasan, A. Bhattacharjee, K. Samin, M. Hasan, M. Basak, M. S. Rahman, and R. Shahriyar, "Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Cedarville, OH, USA: Association for Computational Linguistics, Nov. 2020, pp. 2612–2623. [Online]. Available: https://aclanthology.org/2020.emnlp-main.207

[67] G. Remmiya Devi, P. Veena, M. Anand Kumar, and K. Soman, "AMRITA-CEN@ FIRE 2016: Code-mix entity extraction for Hindi-English and Tamil-English tweets," in *Proc. CEUR Workshop*, vol. 1737, 2016, pp. 304–308.

[68] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, and J. P. McCrae, "Corpus creation for sentiment analysis in code-mixed Tamil-English text," in *Proc. 1st Joint Workshop Spoken Lang. Technol. Under-Resour. Lang. (SLTU) Collaboration Comput. Under-Resour. Lang. (CCURL)*. Paris, France: European Language Resources Association, 2020, pp. 202–210.

[69] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, and M. Wojatzki, "Measuring the reliability of hate speech annotations: The case of the European refugee crisis," 2017, *arXiv:1701.08118*.

[70] P. Fortuna, J. Rocha da Silva, J. Soler-Company, L. Wanner, and S. Nunes, "A hierarchically-labeled Portuguese hate speech dataset," in *Proc. 3rd Workshop Abusive Lang. Online*, 2019, pp. 94–104.
[71] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, Apr. 1960.
[72] Z. Harris and Z. S. Harri, "Distributional hypothesis," *Word World*, vol. 10, no. 23, pp. 146–162, 1954.
[73] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.
[74] H. Tayyar Madabushi, E. Kochkina, and M. Castelle, "Cost-sensitive BERT for generalisable sentence classification on imbalanced data," in *Proc. 2nd Workshop Natural Lang. Process. Internet Freedom: Censorship, Disinf., Propaganda*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 125–134.
[75] X. Ma, P. Xu, Z. Wang, R. Nallapati, and B. Xiang, "Domain adaptation with BERT-based domain classification and data selection," in *Proc. 2nd Workshop Deep Learn. Approaches Low-Resour. NLP (DeepLo)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 76–83.
[76] A. Cohan, I. Beltagy, D. King, B. Dalvi, and D. Weld, "Pretrained language models for sequential sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 3693–3699.
[77] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=rkYTTf-AZ
[78] A. Abdaoui, C. Pradel, and G. Sigel, "Load what you need: Smaller versions of mutililingual BERT," in *Proc. SustaiNLP, Workshop Simple Efficient Natural Lang. Process.* Cedarville, OH, USA: Association for Computational Linguistics, Nov. 2020, pp. 119–123. [Online]. Available: https://aclanthology.org/2020.sustainlp-1.16
[79] L. van der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, Jan. 2014.

MOHAMMAD TAREQ received the M.B.A. degree from Tsukuba University, Japan, and the Ph.D. degree in accounting from RMIT University, Australia. He is currently the Director of the M.B.A. Program, Department of Accounting and Information Systems, University of Dhaka, Bangladesh. He is a Japanese Government's Prestigious Monobusho and an Australian Government's IPRS Endeavor Scholar. He has published articles in *International Journal of Accounting and Information Management*, *Pacific-Basin Finance Journal*, *Japanese Journal of Administrative Science*, and *Japanese Psychological Research*. His research interests include business and technology, capital market, and governance.

MD. FOKHRUL ISLAM received the bachelor's degree from the Department of Robotics and Mechatronics Engineering, University of Dhaka, Dhaka, Bangladesh, where he is currently pursuing the master's degree. His research interests include graph representation learning, natural language processing, computational linguistics, meta learning, and knowledge representation learning.

SWAKSHAR DEB received the bachelor's degree in robotics and mechatronics engineering from the University of Dhaka. He is currently pursuing the master's degree. His current research interests include natural language processing, cognitive science, sentiment analysis, causal inference, multimodal analysis, personalized service, generative model, and reinforcement learning.

SEJUTI RAHMAN received the Ph.D. degree in computer science and engineering from The Australian National University, Australia. She was a Postdoctoral Researcher with The Robotics Institute, Carnegie Mellon University, USA, and the Centre for Autonomous Systems, University of Technology Sydney, Australia. She is currently an Associate Professor with the Department of Robotics and Mechatronics Engineering, University of Dhaka. Her research interests include computer vision and artificial intelligence, with a special emphasis on medical robotics, 3D reconstruction, object recognition, reflectance modeling and illumination, and hyperspectral imaging.

ABDULLAH AL MAHMUD received the M.Sc. degree in international economics and finance, the M.B.A. degree in finance, and the Ph.D. degree in international economics and finance from Brandeis University, USA. He is currently the Director of the M.B.A. Program, Department of Banking and Insurance, University of Dhaka, Bangladesh. He is a Fulbright Scholar.