Language lexicons for Hindi-English multilingual text processing

Mohd Zeeshan Ansari[1], Tanvir Ahmad[1], Mirza Mohd Sufyan Beg[2], Noaima Bari[3]
[1]Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India
[2]Department of Computer Engineering, Aligarh Muslim University, Aligarh, India
[3]Department of Electrical Engineering, Jamia Millia Islamia, New Delhi, India

Article Info

Article history:

Received Aug 11, 2021
Revised Jan 27, 2022
Accepted Feb 12, 2022

Keywords:

Information extraction
Language identification
Language processing
Lexicons
Mixed lingual text

ABSTRACT

Language identification (LI) in textual documents is the process of automatically detecting the language contained in a document based on its content. The present language identification techniques presume that a document contains text in one of the fixed set of languages. However, this presumption is incorrect when dealing with multilingual document which includes content in more than one possible language. Due to the unavailability of standard corpora for Hindi-English mixed lingual language processing tasks, we propose the language lexicons, a novel kind of lexical database that augments several bilingual language processing tasks. These lexicons are built by learning classifiers over English and transliterated Hindi vocabulary. The designed lexicons possess condensed quantitative characteristics which reflect their linguistic strength in respect of Hindi and English language. On evaluating the lexicons, it is observed that words of the same language tend to cluster together and are separable over language classes. On comparing the classifier performance with existing works, the proposed lexicon models exhibit the better performance.

Corresponding Author:

Mohd Zeeshan Ansari
Department of Computer Engineering, Jamia Millia Islamia
Maulana Mohammad Ali Jauhar Marg, New Delhi 110025, India
Email: mzansari@jmi.ac.in

1. INTRODUCTION

Language identification (LI) is applicable to all forms of language, comprising spoken, sign, and handwritten, and pertinent to any form of information storage that uses language. The capacity to reliably recognize the language expressed in a document is an assistive technology that improves information accessibility and has a broad range of applications. Text processing approaches created in natural language processing and information extraction presume that the language of input text is known, and many approaches presume that all documents belong to the same language. Automatic language identification is used to ensure that only text in appropriate languages is submitted for further processing when applying text processing methods to real-world data. LI is required for document collections when the languages of documents are unknown a priori, such as data scraped from the internet, in order to prepare the multilingual index of the document collection by the language in which they were authored. The identification of a document language for routing to an appropriate translation is another use of LI that precedes computational approaches. LI has gained prominence as a result of the emergence of machine translation, which needs the source language of the text to be determined first. LI significantly helps with documentation and usage of low-resource languages.

LI is an old problem in natural language processing (NLP) that has been widely studied both in the speech and text domains [1], [2]. The task of LI in mixed lingual documents is defined as the automatic detection of language(s) present in the document at sentence, phrase or word level based on the content of the document [3]. Many existing LI techniques presume a document to contain text only in a single language from a given set of languages. However, this presumption stands untrue while dealing with multilingual documents, especially those found on the web, that contain text from more than one language from the candidate set. Further, most of the NLP systems assume input data to be monolingual in nature and by the inclusion of data from foreign languages in such systems, noise is introduced and performance degrades [4]. One of the major challenges in LI in multilingual documents stems from a shortage of labelled multilingual text for training the LI models. Standard corpora of multilingual documents are rarely available, whereas corpora of monolingual documents are readily available, even for a reasonably large number of languages [5]. Lexical databases such as Wordnet and BabelNet are collections of entries, each of which contains hierarchical text that provides information on a particular concept. The bulk of such lexical databases, including conventional dictionaries, are relational in form and entirely textual in content. Their organizational structure does not reflect the quantitative nature of words. To address this issue, we created a Hindi-English dataset, with minimal human intervention, by integrating different monolingual language corpora and subsequently, produced the lexicons with language strength associated with them.

The present language models are capable of capturing sufficient semantic information, however, they fail to differentiate the language present in the text. Moreover, morphological characteristics are explicitly extracted as features from the text for language identification. It is therefore important to design the models that can condense the linguistic strength of each word and represent them with interpretable lexicons. The language lexicons presented in this work augment the language models with additional compact information which helps to discriminate between texts in separate languages. We develop domain-independent Hindi-English language lexicons utilizing monolingual corpora that enhance the language-aware learning models capable of performing a variety of language processing tasks such as information extraction, and sentiment analysis. The proposed lexicons are not only focused on modelling a few particular linguistic characters, but rather modelled from a broader view of the lexicon as a key component of language strength. Secondly, they feature a very straightforward, flat structure that does not impose any ordered or hierarchical structure on the vocabulary. Thirdly, they deal with the out of vocabulary problem due to its inherently coupled character level methodological design.

Many existing research works on document-level language identification consider only mono-lingual documents. However, the task of LI is far from solved, particularly, when dealing with multilingual documents, short texts and informal styles such as those found in the real world and social media platforms, or while working with language pairs which are closely related [6]–[8]. Many earlier works in the text domain have utilized word or character n-gram features followed by linear classifiers [9]. A bleak picture depicting support for low-resource languages in automatic language identification has been painted by Hughes *et al.* [3]. LI in multilingual documents is performed using a generative mixture model combined with a document representation. The model learns a language identifier for multilingual documents from monolingual training data which is more abundant as compared to labelled multilingual textual data [10]. Word-level language identification was largely addressed using supervised techniques. For example, King and Abney show that the problem can be framed as a sequence labelling problem and that using hidden Markov models (HMMs) and conditional random fields (CRFs) [11]. The problem can be trained to perform reasonably well at labelling words in multilingual texts starting with monolingual data. In the first shared task on LI on code-switched data in 2014, system designs varied from rule-based systems to those that used word embeddings, enhanced Markov models, and CRF autoencoders [12]. While most teams focused on multilingual LI systems for the shared task, there are approaches that specifically deal with classification on bilingual code-switched texts. For example, Suleep *et al.* built a system that uses several heuristic features, including a special edit distance between Hindi and English that fits their use case for “Hinglish” texts [13].

Unsupervised techniques to language identification at the word level have not proven as popular [11]. Rabinovich and Wintner used a cluster-and-label strategy to discover unsupervised “Translationese” for machine translation, but only for text passages of 2000 tokens [14]. The cluster labeling methodology is another feature that distinguishes their technique from ours. Their automated labelling method creates representative language models for the class labels, which are subsequently assigned to the unsupervised clusters by comparing them to the clusters empirical distributions. We resorted to manual labeling since their methodology is not appropriate in our instance because tagging clusters for our purpose involves very little work.

Recently, some researchers have utilized models based on artificial neural networks in addition to the usual machine learning techniques [15]–[19]. For the SPA-ENG and Nepali-English datasets from the first shared task on language identification in code-switched data, Chang and Lin employ an recurrent neural network (RNN) architecture with pre-trained word2vec embeddings [20]. For the Spanish-English and modern standard Araic-Dailectal Arabic datasets from the second shared task on language identification in code-switched data, Samih *et al.* [21] built a long short-term memory network (LSTM)-based neural network architecture. Their model blends pretrained word2vec embeddings with word and character representations [21]. Using a Hindi-English code-mixed speech corpus from student interviews, Dey and Fung examined the grammatical contexts and motives. Many researchers have looked at the detection of code-mixing in text [22]. The prediction of the places in Spanish-English phrases when the orator transitions between the languages was started by Solorio and Liu [23]. In [24]–[26] investigates code-mixing in brief messages and information retrieval queries. Jamatia *et al.* utilized various linguistic models, dictionaries, and probabilistic models such as conditional random fields (CRFs) and logistic regression (LR) to experiment on Turkish and Dutch forum data [27].

2. PROPOSED METHOD

For the preparation of the Hindi-English dataset, two separate rich language sources are utilized. The Hindi words are taken from the Hindi transliteration dataset given by [28] consisting of 30,696 Hindi words written in their transliterated form in the Roman script. The dataset was manually filtered by 2 annotators having Hindi as their native language in order to remove incorrectly presented words such as “everybody”, “sing”, and “something”. That do not belong to the Hindi language, hence leading to a disagreement score of 0.4% over the unknown words. Further, after the removal of duplicate words, trailing spaces, and new lines, 25,640 unique Hindi words were obtained that are annotated as Hi thus forming the Hindi words dataset. English words are taken from the frequent word list of the British National Corpus and are annotated as En labels. Finally, both the English and Hindi datasets were combined into one single one consisting of 36,429 words, appropriately annotated into 2 classes-Hindi (Hi) and English (En) and randomly shuffled. The final dataset contained some drgree of class imbalance, with English and Hindi words comprising about 29.62% and 70.38% of the dataset, respectively. Table 1 summarizes the distribution of Hindi-English language tags in our dataset and the word lengths of the prepared dataset. It is observed that there is indeed a high fraction of Hindi tags.

The lexicon dataset is used for building and evaluating an automatic LI model. We tokenize the words at the character level and train a Bi-directional long short term memory (Bi-LSTM) classifier with softmax classification. The softmax output for Hindi Tags is considered as the score for language strength. We also present a set of n-gram features using which the logistic regression learns to predict the language tag of a token and, subsequently, generates the second score.

2.1. Learning the classifiers

2.1.1. LSTM

The recurrent neural networks (RNNs), are a kind of neural network that operates with sequential input. They accept a series of vectors as input and output a new sequence that provides information about the sequence at each step in the input. Although RNNs are capable of learning lengthy dependencies in principle, they do not do so in reality and are biased towards the most recent inputs in the sequence [29]. Long short-term memory networks (LSTMs) have been found to capture long-range dependencies and have been built to overcome this problem by integrating a memory-cell. They accomplish so by controlling the quantity of the input delivered to the memory cell, as well as the percentage of the previous state to forget, utilizing several regulatory gates [30]. The LSTM computes a representation of the left context of a character sequence having n words, each represented as a d-dimensional vector. Developing a representation of the appropriate directions is accomplished by reading the same sequence in reverse using a second LSTM. The former will be referred to as the forward LSTM, while the latter will be referred to as the backward LSTM. These are two separate networks, each with its own unique set of parameters. A Bi-LSTM is a pair of forward and backward LSTMs. A word is represented in this model by concatenating its left and right context representations. These representations effectively incorporate a contextual representation of a word, which is useful for a variety of applications.

Table 1. Data statistics for Hindi-English lexicon dataset

	#counts	%age	Max Word Length	Average Word Length	Example Words
English	25640	70.38	20	7.63	*public, deception, great, synchronous, convinced, dramatically*
Hindi	10789	29.62	15	6.81	*gulaam, khiladii, paayal, samudr, himmatawala, mangaladaata*

2.1.2. Logistic regression

It is a popular method for examining and defining a connection between a binary response variable and a collection of input variable. Logistic regression is a linear classifier that fits data to the logistic function and predicts the probability of an event occurring. In order to train the classifier, we utilize the N-grams langaiuge models for N=1-5 as features.

2.2. Language score

The aim is to classify the words into their language and subsequently generate a probabilistic score that denotes the language strength of the words. This strength is a quantative measure of Hindi-ness in a word, which can be a multidimensional vector, each element having a value between 0 and 1. The element value close to 1 means that the chances of word to be Hindi are high. On the other hand, if it is close to zero, the chances of word being Hindi are low. Since we have only two language pairs, Hindi and English, the values being close to one also signifies that chances of a word being English are high. Subsequently, we call the word along with its language strength vector as the language lexicon. We chose to prepare language lexicons of the bidimensional language strength vector by generating two scores using the LSTM and logistic regression classifiers separately. The problem of generating such scores is considered binary classification, however, instead of predicting the class label, we utilize the classifiers to predict the class probabilities.

2.2.1. Score 1

Given a word as a sequence of characters, $x_1,\ldots,x_T$, we learn the classifiers to generate the probability $p(y|x)$ against the true labels y. Here T is the length of sequence, the value of which in our case is 20. The words having length less than T are padded with the NULL character. A fixed-size character level vector representation of each word is generated by applying neural embeddings. The Bi-LSTM over the sequence of character embeddings is applied and the two final hidden states from the front and backward LSTM are concatenated, the output of which is fed into the fully connected network as shown in Figure 1. The output of the fully connected network is then utilized to create the final representation for each input word. The model produces a probabilistic score using the most salient characteristics of the word using this approach. Score 1 is calculated according to the following softmax function as

$$score(x) = \frac{e^{O_j}}{\sum_k e^{O_k}} \tag{1}$$

where $O_j = BiLSTM(X)$ is the output from the last layer of the Bi-LSTM architecture that corresponds to the Hindi language tag.

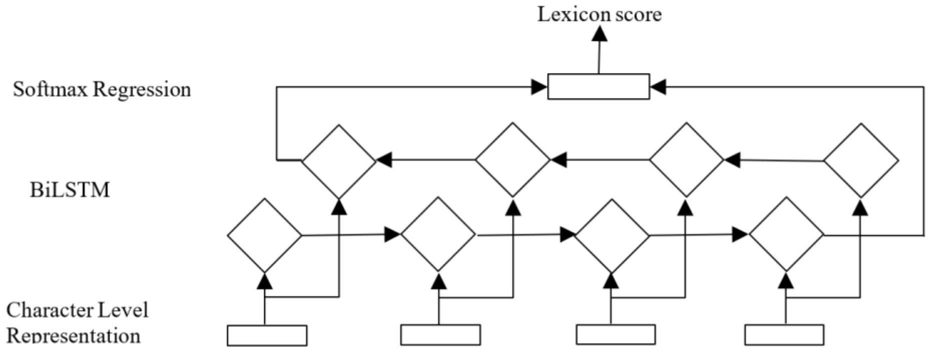

Figure 1. Bi-LSTM network for lexicon score generation

2.2.2. Score 2

For a given word token, the n-gram features are extracted for the values of n=1-5. The n-grams are trained on the logistic regression classifier with default parameters. Apart from predicting the true label y, the probality $p(y|x)$ is calculated. This probability is termed the score 2, which is calculated according to the following logistic function as

$$score(x) = \frac{e^{\theta_0 + \theta_1 X_1 + \ldots + \theta_p X_p}}{1 + e^{\theta_0 + \theta_1 X_1 + \ldots + \theta_p X_p}} \tag{2}$$

where $X_1 + X_2 + \ ... \ + X_p$ is the input n-gram feature set and $\theta_1 + \theta_2 + \ ... \ + \theta_p$ are the parameters learned from the dataset. The score 1 and score 2 calculated from (1) and (2) represent the Hindi language strength of the words present in the lexicons. The score denotes a value between 0 and 1. The score of a word close to 1 is considered a lexicon with very high Hindi language strength and vice-versa.

3. RESULTS AND DISCUSSION

For the experiments, we hold out 10% of the data as a test set. We perform parameter tuning over 10% validation set and report the performance of our models on the test set in Table 2. The reported results are the best among the several runs of models overs different random samples of training dataset. We also compare the performance of our models and illustrate the effect of obtained lexicons by showing the boxplots of language strength scores and by visualizing their tags in a two-dimensional scatter plot.

Table 2. Performance of classifiers on lexicon dataset

		Precision		Recall		F-Score	
Model		LSTM	LR	LSTM	LR	LSTM	LR
English	2697	92.61	95.74	88.25	91.77	90.37	93.71
Hindi	6411	95.15	96.60	97.04	98.28	96.08	97.43
Weighted Avg	9108	94.40	96.34	94.43	96.35	94.39	96.33

3.1. Classifier performance

The results of the investigation in Table 2 reveal that when the models are trained with 27,321 training instances, the overall performance of the classifiers with 9,108 test samples is significant. When the precision, recall, and F1-measure of each class are compared, it is discovered that the predictions of the Hindi class are more accurate than those of the English class, though with a small but significant difference. However, the performance of the classifiers demonstrates that the logistic regression model outperforms the Bi-LSTM model in terms of all the measures. Although the Logistic regression produces the highest F-Score of 96.33, its recall is pretty similar, with a difference of 1.24% only with Bi-LSTM.

3.2. Lexicon analysis

The analysis of the suggested lexicons language strength is carried out by examining the scatter plot and box plot of the scores obtained from (1) and (2) as presented in Figure 2. The scatter plot displays the score 1 on the x-axis and score 2 on the y-axis with the blue data points representing English class labels and the orange data points representing Hindi class labels. It demo nstrates that the lexicons are clearly separable with small quantity of outliers, hence, when the various Hindi-English multilingual datasets are supplemented with these lexicons, the models would perform better, which is consistent with our proposed concept of language lexicons.

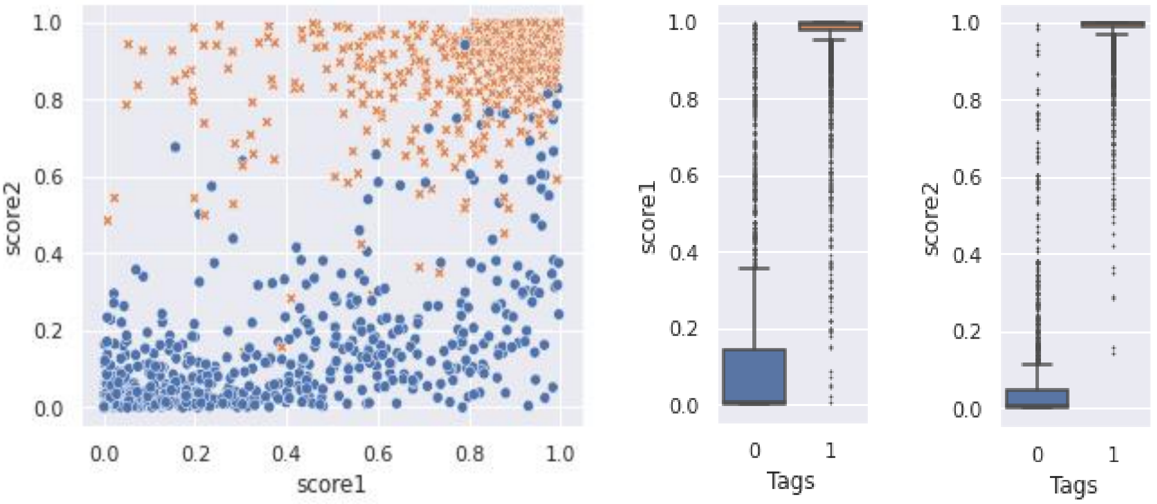

Figure 2. Scatter plot and box plot of lexicons

The box plots of Figure 2 reveal that the scores of Hindi words (denoted by tag 0) have very high language strength being majority of scores very close to 1. On the other hand, English words have very low Hindi language strength, with a majority scores of less than 0.2. Table 3 shows some sample lexicons generated by the proposed methodology, including the outliers. We see the disagreements between the two scores in a small number of lexicons, which is also observed from the box plots. The comparative picture of proposed and previous similar methods, in Table 4, shows that the performance of our approach is fairly better than other approaches and obtains the highest average F1-measures in both the proposed models.

Table 3. Sample language lexicons

Word	2-dimensional Hindi Language strength	
abhilaasha	0.9943395256996155	0.9999969538347858
baharoon	0.9996402263641357	0.9969923646323342
khuski	0.9981417655944824	0.9987074582327405
literally	0.9714275598526001	0.2044318907744394
jurisdictional	0.7277640104293823	0.1737373891892225

Table 4. Comparative F1-measures of proposed and previous models

Model	English	Hindi	Average
Lexicon Logistic Regression (proposed)	0.937	0.974	0.963
Lexicon LSTM (proposed)	0.904	0.960	0.943
Veena *et al*. [15]	0.658	0.829	0.804
Bhattu *et al*. [19]	0.831	0.613	0.769
Sequiera *et al*. [17]	0.911	0.651	0.767
Shekhar *et al*. [16]	0.857	0.939	0.742

4. CONCLUSION

The considerable size of proposed language lexicons may be regarded as comprehensive in terms of the number of entries with two-dimensional quantifiable language strength to indicate the extent of the possibility that a word belongs to the Hindi or English language. These language lexicons were created by utilizing the complementary English and Hindi Roman vocabulary. After the extraction of significant features from the vocabulary, logistic regression and Bi-LSTM classifiers are trained to obtain the probabilistic scores. The scores resemble the Hindi linguistic power of each word in a two-dimensional space. The study of lexicons acquired shows that they have acquired language characteristics such that they have high values for Hindi words and low values for English words. The comparison of classifier performance with previous recent work shows that the proposed methods outperform the previous methods. Additionally, the proposed models may be used to assess the language strength of new words and may be integrated with any kind of multilingual model. The proposed lexicon models may be significant in a variety of tasks pertaining to code-mixed text such as language identification, sentiment analysis, and information extraction. By fine tuning the models over the application dataset, domain specific lexicons can be generated and utilized.

REFERENCES

[1] A. S. House and E. P. Neuburg, "Toward automatic identification of the language of an utterance. I. Preliminary methodological considerations," *The Journal of the Acoustical Society of America*, vol. 62, no. 3, pp. 708–713, Sep. 1977, doi: 10.1121/1.381582.

[2] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," 1994.

[3] C. E. Hughes, E. S. Shaunessy, A. R. Brice, M. A. Ratliff, and P. A. McHatton, "Code switching among bilingual and limited English proficient students: possible indicators of giftedness," *Journal for the Education of the Gifted*, vol. 30, no. 1, pp. 7–28, Sep. 2006, doi: 10.1177/016235320603000102.

[4] T. Baldwin, P. Cook, M. Lui, A. MacKinlay, and L. Wang, "How noisy social media text, how diffrnt social media sources?," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 356–364.

[5] M. Lui and T. Baldwin, "Cross-domain feature selection for language identification," in *Proceedings of 5th International Joint Conference on Natural Language Processing*, 2011, pp. 553–561.

[6] S. Gella, K. Bali, and M. Choudhury, "'Ye word kis lang ka hai bhai?' Testing the limits of word level language identification," in *Proceedings of the 11th International Conference on Natural Language Processing*, 2014, pp. 368–377.

[7] P. Wang, N. Bojja, and S. Kannan, "A language detection system for short chats in mobile games," 2015, doi: 10.3115/v1/w15-1703.

[8] M. Z. Ansari, M. M. S. Beg, T. Ahmad, M. J. Khan, and G. Wasim, "Language identification of Hindi-English tweets using code-mixed BERT," Jul. 2021, [Online]. Available: http://arxiv.org/abs/2107.01202.

[9] M. Z. Ansari, M. B. Aziz, M. O. Siddiqui, H. Mehra, and K. P. Singh, "Analysis of political sentiment orientations on twitter," *Procedia Computer Science*, vol. 167, pp. 1821–1828, 2020, doi: 10.1016/j.procs.2020.03.201.

[10] M. Lui, J. H. Lau, and T. Baldwin, "Automatic detection and language identification of multilingual documents," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 27–40, Dec. 2014, doi: 10.1162/tacl_a_00163.

[11] B. King and S. Abney, “Labeling the languages of words in mixed-language documents using weakly supervised methods,” in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 1110–1119.

[12] T. Solorio *et al.*, “Overview for the first shared task on language identification in code-switched data,” 2014, doi: 10.3115/v1/w14-3907.

[13] H. J. Suleep, K. B. Vaskar, and Raychoudhury, “Word-level language identification in bi-lingual code-switched texts,” in *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*, 2014, pp. 348–357.

[14] E. Rabinovich and S. Wintner, “Unsupervised identification of translationese,” *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 419–432, Dec. 2015, doi: 10.1162/tacl_a_00148.

[15] P. V Veena, M. Anand Kumar, and K. P. Soman, “Character embedding for language identification in Hindi-English code-mixed social media text,” *Computación y Sistemas*, vol. 22, no. 1, Mar. 2018, doi: 10.13053/cys-22-1-2775.

[16] S. Shekhar, D. K. Sharma, and M. M. S. Beg, “Language identification framework in code-mixed social media text based on quantum LSTM-the word belongs to which language?,” *Modern Physics Letters B*, vol. 34, no. 06, p. 2050086, Feb. 2020, doi: 10.1142/S0217984920500864.

[17] R. Sequiera *et al.*, “Overview of FIRE-2015 shared task on mixed script information retrieval.” pp. 19–25, 2015.

[18] A. Das and B. Gambäck, “Code-mixing in social media text: the last language identification frontier?,” *Revue TAL*, vol. 54, no. 3, pp. 41–64, 2015.

[19] S. N. Bhattu and V. Ravi, “Language identification in mixed script social media text.” pp. 37–39, 2015.

[20] J. C. Chang and C.-C. Lin, “Recurrent-neural-network for language detection on twitter code-switching corpus,” Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.4314.

[21] Y. Samih, S. Maharjan, M. Attia, L. Kallmeyer, and T. Solorio, “Multilingual code-switching identification via LSTM recurrent neural networks,” in *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, 2016, pp. 50–59, doi: 10.18653/v1/W16-5806.

[22] A. Dey and P. Fung, “A Hindi-English code-switching corpus,” 2014.

[23] T. Solorio and Y. Liu, “Part-of-speech tagging for English-Spanish code-switched text,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing-EMNLP ’08*, 2008, p. 1051, doi: 10.3115/1613715.1613852.

[24] T. Gottron and N. Lipka, “A comparison of language identification approaches on short, query-style texts,” in *European Conference on Information Retrieval*, 2010, pp. 611–614.

[25] P.-J. Farrugia, “TTS pre-processing issues for mixed language support,” 2004.

[26] M. Rosner and P.-J. Farrugia, “A tagging algorithm for mixed language identification in a noisy domain.” pp. 190–193, 2007.

[27] A. Jamatia, B. Gambäck, and A. Das, “Part-of-speech tagging for code-mixed English-Hindi twitter and facebook chat messages,” in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, 2015, pp. 239–248.

[28] M. M. Khapra, A. Ramanathan, A. Kunchukuttan, K. Visweswariah, and P. Bhattacharyya, “When transliteration met crowdsourcing: an empirical study of transliteration via crowdsourcing using efficient, non-redundant and fair quality control.” pp. 196–202, 2014.

[29] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.

[30] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

BIOGRAPHIES OF AUTHORS

Mohammad Zeeshan Ansari is Assistant Professor at Department of Computer Engineering, Jamia Millia Islamia (A Central University), New Delhi. He received M.Tech in Computer Science and Engineering from Delhi Technological University, New Delhi and B.Tech in Computer Science and Engineering from Uttar Pradesh Technical University, Lucknow. His area of research is Code Mixing, Information Extraction, Text Mining, Natural Language Processing and Deep Learning. His research currently focusses on developing multilingual natural language processing and information extraction models for social media text using with emphasis on supervised learning. He has active participation in several national and international workshops, seminars and published articles in reputed conferences and refereed journals. He is also reviewer of refereed journals. He can be contacted at email: mzansari@jmi.ac.in.

Dr. Tanvir Ahmad is currently Professor in the Department of Computer Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi. He received his Engineering degree from Bangalore University with First Class, M.Tech from I.P. University, New Delhi and Ph.D. from Jamia Millia Islamia, New Delhi. He has more than twenty years of academic, research, training and administrative experiences in the field of Computer Engineering. His areas of research are Text Mining, Graph Mining, Big Data Analytics, and Natural Language Processing. He can be contacted at email: tahmad2@jmi.ac.in

Prof. M. M. Sufyan Beg obtained his B.Tech. (Electronics) degree from the Aligarh Muslim University, India in 1992 with First Rank. He obtained his M.Tech. (Microelectronics) degree from IIT Kanpur, India in 1994. Thereafter, he joined the Department of Electronics Engineering, Aligarh Muslim University, India as a member of the faculty. He has been a Lecturer, a Senior Lecturer and then a Reader in the Department of Computer Engineering at the same University. While on study leave from there, he obtained his Ph.D. degree in the area of Computer Technology from IIT Delhi, India in 2004. He also visited the University of California at Berkeley as a BT Fellow. Currently, he is a Professor and Chairman at the Department of Computer Engineering at the Aligarh Muslim University. He is also serving as the Principal of Z. H. College of Engineering and Technology since August 2015. He can be contacted at email: mmsbeg@hotmail.com

Noaima Bari completed her bachelor's degree in electrical engineering from Jamia Millia Islamia University, New Delhi in 2021 and is currently pursuing her masters in electrical and computer engineering at Georgia Institute of Technology in Atlanta, US. She is interested in the field of machine learning and its applications in language and vision. She wishes to study about optimization of machine learning algorithms and how they can be used to ease tasks in daily lives. She has been a student member of IEEE since 2017. She can be contacted at email: noaimabari@gmail.com.