Qeios

Detection of Language from Roman Urdu and English Multilingual Corpus

Syed Immamul Ansarullah[1], Sajadul Hassan Kumhar[2*], Sami Alshmrany[3]

[1] Department of Computer Applications, Govt. Degree College Sumbal, Bandipora, J&K, India
[2] Research Scholar, SSSUTM, Sehore India
[3] Faculty of Computer and Information Systems, University of Madinah, Medina 42351, Saudi Arabia

* Correspondence: syedansr@gmail.com; sajadulhassan@kwintech-rlabs.org

Abstract

Purpose: This study aims to suggest and validate a model to identify the languages from Roman Urdu and English mixed multilingual corpus collected from social media sites.

Background: The problem of identifying languages from a corpus of written texts that includes two or more languages is known as language identification or detection. Identifying or detecting the language present in social media text is a requirement and it has numerous applications in natural language processing and computational linguistics, like for word embedding generation, emotion analysis and part of speech tagging etc.

Methodology: The dictionary-based baseline with SVM and Bi-Directional LSTM has been used in language identification from collected Roman Urdu and English multilingual Corpus. This research work will help in identify the languages from Roman Urdu and English Corpus. The English and Roman Urdu corpus had been obtained from different social media websites and cross-media platforms such as Facebook, Twitter, Google+, Instagram, WhatsApp, and Messenger, etc. The dictionary-based baseline with SVM and Bi-Directional LSTM has been used in language identification from collected Roman Urdu and English multilingual Corpus.

Results: Based on the results achieved using the methodology in the research work the Bi-directional LSTM model performed better with an accuracy of 97.98%.

Conclusion: The problem in recognizing or detecting the language present in a given document or statement is referred to as language recognition or detection The Corpus of English and Roman Urdu

https://doi.org/10.32388/LI3DN2

is collected from social media websites. The text for training is submitted to a bi-direction LSTM accordingly to verify if the text is in English language or Urdu language. The results of word recognition for bidirectional word-level LSTM from Roman Urdu and English showed improved results.

Keywords: Language Identification, Language Detection, Social Media, Bi-Directional LSTM, Roman Urdu, English, Corpus.

1. Introduction

The problem in recognizing or detecting the language present in a given document or statement is referred to as language recognition or detection. The need for suitable methods for defining the language in an utterance of text or paper is obvious and multifaceted. For example, delivering the web and social media content in the user's native language is an important factor to motivate and attract the user and visitors say A. Kralisch and T. Mandl [1] in 2006. Several Natural Language Processing (NLP) programs and Information Retrieval (IR) systems were built on the assumption that the input language is prescribed or well known. In those circumstances and conditions, a considerable and paramount set of language identification or detection ensures that only a relevant set of particular languages is passed as input to these NLP and IR (Information Retrieval) systems. For Machine translation and transliteration, language detection is the prerequisite of the system for translation and transliterating from one document language to another.

Language identification or detection has been investigated both statistically and linguistically. S. Johnson in 1993 [27] had investigated language detection linguistically and statistically. M. Damashek [18] in 1995 and T. Dunning [30] in 1994 has analyzed the text statistically and marked the language identification as a text categorization task along with W. B. Cavnar et al [32], and D. Elworthy [8] in 1999 [35]. T. Dunning [30] in 1994, M. Damashek [18] in 1995, J. M. Prager [14] in

1999, and P. McNamee [22] in 2005 has carried the language detection in the utterance of text on document level, the sub-document level has been carried out by H. Yamaguchi and K. TanakaIshii [13] in 2012, B. King and S. Abney [4] in 2013. The language detection and identification on sentence-level and word-level have been carried out by D. Nguyen and A. S. Dogruoz [7] in 2013, T. Solorio and Y. Liu [126] [31] in 2008. The assignment of identifying a specific and distinctive language to a text is known as document-level language identification. The better accuracy in the document level language detection has been achieved with formal content D. Nguyen and A. S. Dogruoz [7] in 2013; e.g. in news articles and Wikipedia. The document level detection and identification of language are better and easy compared to other types of language identification such as sentence level and word level monolingual documents. B. King and S. Abney [4] in 2013 stated that the sub-level document identification is implemented and conducted on multilingual documents which are monotonous, difficult, and boring than document-level language detection for new articles, news forms, and blog posts. According to T. Baldwin and M. Lui [29] in 2010 the sub-document level is more finely grained and is carried out on short text quires. The least and most important identification of languages is the word level language detection performed on mixed multilingual content and has been recently investigated by different scholars, social scientists, computer linguistics, and by NLP researchers T. Solorio and Y. Liu (2008) [31], R. Sequiera (2015) [23].

The vast majority of research is available on widely spoken and written resource-rich languages such as English B. Hughes et. al (2006) [3], however, there is a growing demand to conduct computer linguistic and NLP research on low resource languages, especially on social media content. The social media contents are noisy and contain some irregularities which exhibit certain social-linguistic phenomena, such as short text, multilingualism, Romanization, and mixing of mixture languages, etc. These factors pose certain challenges in computational linguistics and natural language processing for the identification of languages.

2. Related Research

The language identification on mixed multilingual social media content is done in four manifold forms, they are the document, the short text sometimes called the sub-document level, the word, and sub-word level classification for the identification of languages. One of the most important research work carried out by W. B. Cavnar et al [32] in 1994, states that language identification was treated to be the document classification. In their research work, the n-gram model was used to capture the similarity of the language across by measuring the ranking of the document. The n-gram was popularized by W. B. Cavnar et al [32] and many other researchers enlightened it further and updated it in many ways. T.Dunning [30] in 1994 used the Markov models and Naive Bayesian classifiers by training and finding the patterns by using the n-gram byte words. The news article has been used to investigate, classify and identify the languages at the document level.

G. Grefenstette [12] suggested two methods for European languages in their computer analysis, one focused on a trigram model and the other on short-words, and found that the two methods worked very well and also for word it counts greater than 50 (word count >50), although a trigram-based approach is powerful and performs well for short texts. J. M. Prager [14] presented a technique for determining the similarity in teaching and evaluating languages that relied on a vector space model built on tf-idf. They used the n-gram model to distinguish monolingual documents from 20 different languages and discovered that the model works best for short texts. M. Padro and L. Padro [21] proposed a hybrid model taking the advantage of the Markov model, n-gram model for text categorization, and trigram frequency vector for six languages, however, the only fact is that the Markov model outperformed better than the other two models. L. Grothe [16] conducted another study on text-level word recognition and identification, comparing the short-word type approach, frequent-word type approach, and n-gram type approach. These models, on the other hand, have been optimized for parameters such that their performance is comparable.

In a web paper, T. Baldwin and M. Lui [29] proposed and validated a Supervised machine learning language identification algorithm based on the n-gram model. They discovered that the SVM

and 1-Neural network model with a linear kernel outperform other models. The scale of the training datasets is limited and the duration of the text is brief. However, the model becomes difficult to use when the number of languages is high. Subsequently, in 2011, they proposed a new cross-domain linguistic recognition model, based on Naive Bayesian, undermining discriminatory features in 97 languages and using 5 datasets that would outperform W.B. Cavnar's and J.M. Trenkle's (1994) approach [34]. M. Lui and T. Baldwin [20] suggested langid.py tools in Python on Nave Bayesian and N-grams models. The NLP approach is famous for its usability and robust linguistic identification in the NLP community. The Markov model HMM proposed by A. Xafopoulos et al (2004) [2], URL base web search information model proposed by E. Baykan et al [9] in 2008, and corpus formation for low resource language proposed by K. P. Scannell (2007) are several examples.

Sentence-level language identification is much tough, tedious, and harder than document-level language identification due to short text and much noise such as hashtags, comments, and Twitter retweets, etc. To distinguish tweets in six languages, E. Tromp and M. Pechenizkiy [10] suggested a model based on a graph-based n-gram model. However, the model has not worked for word-level identification. S. Bergsma et al [24] proposed another model for sentence identification of languages in the tweets of Twitter which contains nine languages. However, the model has little investigation for Romanized languages and is considered only for monolingual sentences and tweets. S. H. Kumhar et al [26] conducted a detailed survey on language identification for corpus collection and identification of languages. M. Goldszmidt et al [20] proposed a character frequencies classifier of tweets and build a bootstrapping-based language model for language identification. The Wikipedia training dataset has been used. The model says nothing about Romanized texts. Later S. Carter et al [25] 2013 suggested a character feature model and explored the language present, as well as hashtags, in five languages: Dutch, German, French, Spanish, and English. They worked better in all languages studied, but the model is script-based, and no study on mixed corpora has been studied. K. N. Murthy and G. B. Kumar [15] presented a linear regression language identification model for six Indian languages that were written in short texts. Oriya, Hindi, Marathi, Malayalam, Punjabi, Tamil, Telugu,

Kannada, and Bengali are among the six. However, their words existed in language-specific scripts and did not take into account Romanized texts. Multilingualism's language granularity is no longer limited to the text or sentence stage. Modern social media development has altered the way of traditional writings. Mixed multilingualism is common among social media users. It is evident and worthwhile when multiple and mixed languages are used on social media to study language identification at the word level. Recently more focus received much attention towards the granularity of data at the word level and much work has been published.

T. Solorio and Y. Liu in 2008 [31] proposed a dictionary based on the n-gram model, in which the language switch on spoken languages of Spanish-English has been carried out for identification at the word level. They went on to explore how, by using language knowledge, the monolingual POS tagger can be used, help understand language switching. However, the model hasn't investigated multilingual POS taggers. H. Yamaguchi and K. Tanaka-Ishii (2012) [13] proposed another model for language identification through dynamic programming on an artificial mixed multilingual corpus at the word level. They created the language segments by sampling random texts from the monolingual corpus. B. King and S. Abney [4] proposed a semi-supervised model for language identification on a dataset of 30 languages. They explored the Naïve Bayes classifier for individual language identification by classifying and sequence labeling using CRF generalized expectation criteria.

The Model for Dutch languages at the root level was proposed by D. Nguyen and A. S. Dogruoz [7] in 2013 using tags, dictionaries, logistic regression classification, and CRF for defined Dutch languages. In multilingual languages and conversational data which are typically less distinct from each other or dialects, the proposed study model is not good. C. Lignos and M. Marcus (2013) [5] suggested a model for language recognition based on the ratio of the likelihood of word in language on bi-lingual Spanish-English tweets. However, the current research would not address word uncertainty in lexical similarity. C. R. Voss et al. (2014) [6] suggested a maximum entropy model for Romanized Code-mixed Monolingual Arabic, English, and French. G. Chittaranjan et al

[11] used the CRF hand-craft orthographic and contextual features to execute the script. S. Rijhwani [28] combined the Markov model for language recognition with a variety of code-mixed language semantic materials. The shared task has encouraged word language recognition in recent years; T. Solorio [31] has been working for the first time on text switching computing methods. The joint FIRE mission was followed in 2015 by an additional workshop for code mixed information extraction in resourced languages, R. Sequiera et al. [23], and the second shared codeswitching workshop, M. Diab [19]. Many researchers have researched language comprehension from several different language pairs, including Hindi-English, Spanish-English, and Nepali-English, in these workshops. They proposed and investigated different models for language identification on mixed corpus by using machine learning techniques such as SVM, Supervised learning, logistic regression, CRF, LSTM, etc. More than that research works on handcrafted orthographic and contextual feature extraction mechanisms while folding their algorithms. These handcraft algorithms include n-gram, dictionary, capitalization, Information, presence of numbers, and punctuations in sentences and words.

3. Proposed Method for Language Detection

The research work will concentrate on a dictionary-based baseline for language identification. We proposed a multi-lingual mixed Urdu-English corpus originating from social media pages like Twitter, Facebook as well as blogs. This dictionary is made up of a list of terms from each group. The python script is run on the language structure representation, such as Urdu words are written in Roman script (transliterated Urdu), followed by Nastaliq script, to construct the Urdu dictionary. The following figure 1 is the architecture used with a dictionary-based baseline.

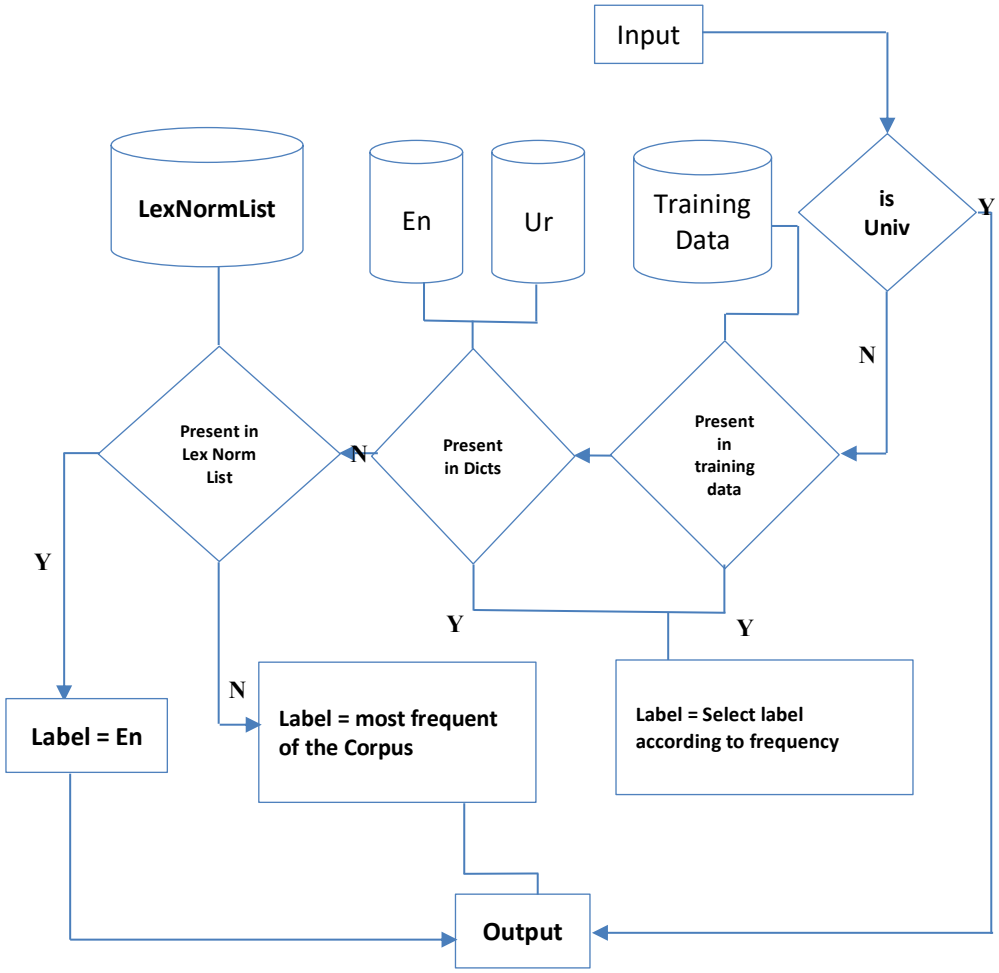

Fig. 1. Showing the dictionary-based system of language detection

In our research work, we will follow this procedure in the method of mark prediction for each token in the related test split, and make appropriate improvements in the field of Roman Urdu.

The training data first pass the 'Uni' module of the above dictionary to predict universal language. If the term identified is in the Uni list, the training outcomes are predicated on the most important mark. We would look at 4/5 of the training outcomes if the token is not 'Uni.

If the assessing token is not present in the course, we should consult Urdu or English dictionaries. The token's presence or normalized frequency in the two dictionaries can be used to estimate its presence.

We would consult LexNormList and the dictionary if the text is missing from the split training results. If the token is in LexNormList, we predict “En,” otherwise we pick the next in the related training split, such as Ur. We can try on different thresholds and frequencies on the above-mentioned

dictionaries. We perform the classification by using the support vector machine with the combination of n-gram, presence of dictionaries, length of word by a decision tree, capitalization, and context of the word.

Following the work of B. King and S. Abney, we used n-gram as a function that has been used by most language recognition researchers [21], we use the length from n= 1 to 5 for n-gram and word as the feature in the study. And then, inclusion has been used as a feature in the dictionary for all of the dictionaries available in the experiment for language recognition. Furthermore, we trained the decision tree with a length of term as multiple features and used the decision tree nodes to create a Boolean algorithm. To add capitalization, three Boolean features were used.

Due to the importance of the CRF model, we used the stochastic gradient descent linear chain SRF and the penalty parameter L1. We used a similar feature to identify support vector machines with a word length of 5 and the prefix and suffix in the n-gram model. The research workers have used the baseline system's dictionary projections instead of a binarized function to provide features for each token of a single dictionary as well as the raw length of the context of the token.

In research, the help vector machine has been used to distinguish data with various combination features. Character n-grams, the inclusion of a dictionary, the length of the expression, capitalization, and hints are some of the features that have been used to characterize the results. The n-gram has been used with character lengths ranging from 1 to 5 and the attribute being the letter. The existence of a dictionary has been used as a feature of the experiment to estimate all possible dictionaries. Using the J48 decision tree, the length of a word generates various length functions. The capitalization function makes use of the three Boolean features to allow capitalization regardless of whether the word is all capital, first letter capital, or small. Since contextual cues play an important role in language recognition, we used the previous and next terms of the token as a function.

At the beginning of the term, in our experiment, we used the dollar sign ($) and the symbol £ as the end. The following table 1 shows a preview of our feature collection.

Table 1. Features created for the word ‘pehla' in a text fragment: ‘woh pehla maira'

Feature Name	Feature Example (with value=1)
G (character n-gram)	$p, e, h, l, a£, $pe, eh, la£, $peh, hela£, $pehla£
D (Dictionary)	<dict-train-Ur>
L (length)	<5-7>
P_1N_1 (Context)	<p1-who>, <n1-maira>

According to C.-W. Hsu et al, SVM Kernel's C parameter is highly optimized and has several functionalities in mind. While operation radial base is more efficient than a linear kernel, C and the computationally costly parameters of large feature sets can only be achieved after base optimization. For each feature set, the optimization parameter c is performed 2-15 to 210. C = 0.0312 showed that the cross-validation was the best and most suited for the GDLCP1N1 run.

Table 2. Average SVM word level validation accuracy

Features	**Accuracy**	**C**
GD	92.45	0.0009
G	91.75	0.0156
GDL	92.61	0.0019
$GDLCP_1N_1$	93.06	0.0312
GDLC	92.59	0.0019

The LSTM RNN performed well in sequence labeling tasks such as language recognition, named object relationship, and POS tagging. Y. Samih et al [33] have recently explored the use of LSTM for mixed corpus. In the research work, the bi-directional LSTM model has been used to identify languages in mixed Urdu-English multilingual results. To complete the task of word-level language recognition, two LSTM architectures are used. The first architecture is simple and uses vectors as input, while the second uses two LSTM combinations, one to identify character levels and the other to identify word levels.

Word level bi-directional LSTM

The word-level bi-directional LSTM model accepts feedback in the form of a character series (w = w1 + w2). For a given word w, the LSTM generates word embeddings from Table 2 for vector representation of $x_t \in R^{d^w}$ which makes the first layer of the model. The LSTM uses the network unit initialization for the $X_{1:T}$ T vector sequence and interactively maps the $X_{1:T}$ sequence in a $Y_{1:T}$ output sequence. The following equation is used interactively.

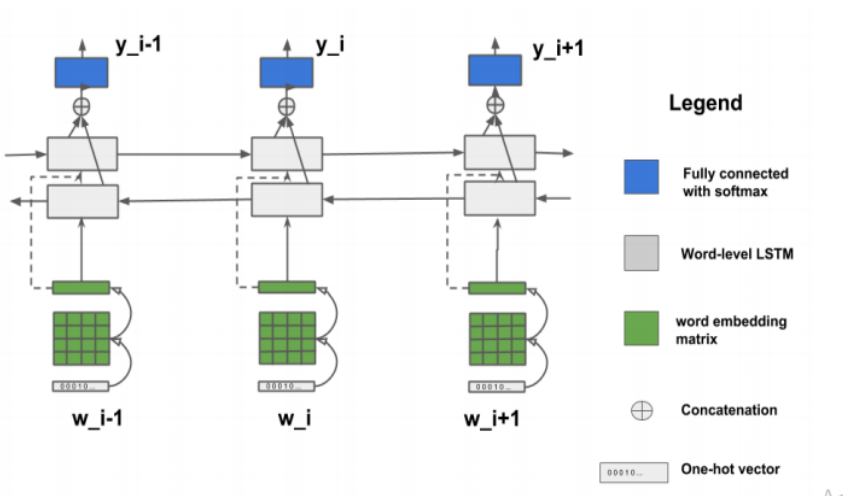

Fig: 2. Word-level LSTM Model for LID

$$i_t = \sigma_g\,(w^x{}_i x_t + w^h{}_i h_{t-1} + w^c{}_i C_{t-1} + b_i$$

$$f_t = \sigma_g\,(w^x{}_i x_t + w^h{}_f h_{t-1} + w^f{}_i c_{t-1} + b_f)$$

$$O_t = \sigma(w^x{}_0 x_t + w^h{}_0 h_{t-1} + w^c{}_0 c_{t-1} + b_0)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ tanh\,(w^x{}_c x_t + w^h{}_c h_{t-1} + b_c)$$

$$h_t = O_t \circ tanh(c_t)$$

At timestamp t, $x_1 \varepsilon R^{d^w}$ is the LSTM input, i_t is an activation vector for the input gate, f_t which acts as input vector to the forged gate, O_t which is output from the activator vector at the output gate, h_t is the hidden vector and c_t is cell state vector in the LSTM. In the LSTM, b_s represents the bias vector, w_s represents the weight matrix and h_t is a logic sigmoid activation curve and tanh represents the hyperbolic tangent function in the entry wise product. The architecture of these two LSTMs was one in the reverse direction and one in the forward direction. Therefore, the h_t is the final output of the concatenation of two vector $h^{forward}$ and $h^{backward}$ given below:

$$\widetilde{h_t} = h_t^{forward} \oplus h_t^{backward}$$

In the preceding equation, $\oplus$ a symbol has been used as a concatenation operator. Finally, a fully connected layer of softmax enabling function is transferred via the concatenation vector to produce y_t:

$$y_t = softmax(W_y \tilde{h}_t + b_y)$$

The architecture is shown in figure 2 above. This network's loss function is the group loss entropy model.

$$L_{cross_{entropy}}(y, \hat{y}) = -\sum y_i log\ (\hat{y}_i)$$

The hyperparameters of the function are showing in table 3 given below

Table 3.: Word-Level LSTM hyperparameters

Hyper Parameter	Valu e
Rate of Learning	0.05
LSTM Hidden Units	254
Rate of Dropping	0.5
Size of the batch	63
No of Epochs	10
d^w (Word Size of Vector)	100

Word and Character Level Bi-directional LSTM

This module takes both the character of words and words as an input vector to the network LSTM. At timestamp t, for a given word w, the character sequence of the word be $c^{wt}{}_{1:M} = \{C^{wt}{}_1, C^{wt}{}_2, \ldots C^{wt}{}_n \ldots C^{wt}{}_M\}$. When one hot of character's representation $C^{wt}{}_m$ passes through the character vector table which generates d^c dimension vector ch_m. The fast Text of skip-gram generates the embeddings for this LSTM. The embedding matrix for w_t generated M x d^c Vector size which is used as a bi-directional LSTM character entry. The LSTM concatenates the final output of both forward and backward bi-directional char-LSTM and encodes them to a fixed representation of $h^c{}_m$. A vector $x_t \epsilon R^{d^w}$ representation is produced for w_i from the word embedding see in table 4 and a vector representation x_t and a vector $h^c{}_M$ is combined to produce v1 for a single presentation of $\bar{w}^h{}_c$ functions. The word series $w_1, w_2, \ldots\ldots w_t$ is a vector representation $v_1, v_2, \ldots\ldots v_T$ that is achieved by concatenation of both forward and reverse direction vectors outputs and is moved to a bi-directional LSTM Char-Level, works on words, and generates outputs as $y_1, y_2 \ldots y_r$.. The architecture of the combined word and Char-Word bidirectional LSTM is shown below. The formal definition of the module follows.

Char-LSTM

$$i_m{}^{char} = \sigma_g\,(w^{ch}{}_i Ch_m + w^h{}_i h_{m-1}{}^{char} + w^c{}_i c_{m-1}{}^{char} + b_i^{char}$$

$$f_m^{char} = \sigma_g\,(W_f^{ch} Ch_m + W_f^h h_{m-1}^{char} + W_f^c c_{m-1}^{char} + b_f^{char})$$

$$O_m^{char} = \sigma(W_\circ^{ch} Ch_m + W_\circ^h h_{m-1}^{char} + W_\circ^c c_{m-1}^{char} + b_\circ^{char})$$

$$c_m^{char} = f_m^{char} \circledcirc c_{m-1}^{char} + i_m^{char} \circledcirc tanh\,(W_c^{ch} ch_m + W_c^h h_{m-1}^{char} + b_c^{char})$$

$$h_m^{char} = o_m^{char} \circledcirc tanh(c_m^{char})$$

In the final step, when m=M and considering the bi-directional case

$$h_M^c = (h_M^{char})^{forward} \oplus (h_M^{char})^{backward}$$

Obtaining single representation v_t for w_t

The word LSTM

$$i_m{}^{word} = \sigma_g\,(W_i^v v_t + W_i^h h_{t-1}^{word} + W_i^c c_{t-1}^{word} + b_i^{word}$$

$$f_t^{word} = \sigma_g\,(W_f^v v_t + W_f^h h_{t-1}^{word} + W_f^c c_{t-1}^{word} + b_f^{word})$$

$$O_t^{word} = \sigma(W_\circ^v v_t + W_\circ^h h_{t-1}^{word} + W_\circ^c c_{t-1}^{word} + b_\circ^{word})$$

$$c_t^{word} = f_t^{word} \circledcirc c_{t-1}^{word} + i_t^{word} \circledcirc tanh\,(W_c^v v_t + W_c^h h_{t-1}^{word} + b_c^{word})$$

$$h_t^{word} = o_t^{word} \circledcirc tanh(c_t^{word})$$

Considering bi-directional case

$$\tilde{h}_t^{word} = (\tilde{h}_t^{word})^{forward} \oplus (\tilde{h}_t^{word})^{backward}$$

And finally, the softmax function becomes.

$$y_t = softmax(W_y h_t^{w\tilde{o}rd} + b_y)$$

Table 4.: Word parameters + LSTM character-level parameters

Hyper Parameter	Valu e
Rate of Learning	0.05
Hidden Unit of LSTM	256
Rate of Dropout	0.5
Size of batch	64
No of Epochs	10
d^w(Word size of vector)	100
d^c(Character Size of Vector)	100

4. Result and Discussion

For the identification of language words present in the dataset the Dictionary, Support Vector Machine, Long Short-Term Memory, and CRF-based tools and methods had been used. Table 5

represents the overall accuracy and labeling for these datasets produced from these systems of tools and methods.

Table 5. Label accuracy for Dictionary, SVM, CRF, and LSTM Systems

Label	Dict.	CRF	SVM	LSTM
En	91.09	91.99	92.00	98.37
Ur	94.99	97.08	95.13	99.26
Mixd	18.10	18.10	51.99	56.71
Ne	24.98	24.83	75.56	78.04
Acro	55.24	52.97	76.34	77.98
Uni	98.13	98.03	99.07	99.11
Overall	90.32	92.29	93.63	97.98

The LSTM Bi-Directive word-level method reaches the maximum level of precision with a value of 97.98% of all available systems with fivefold cross-validation accuracy. To use two side bootstrap Ephron on Graham et al pseudocode, we carried out a statistical significance test. We are using the 1k and α sample test = 0.05, to find the statistically important improvements in the SVM, the CRF, and the LSTM over the dictionary-based method, improvement in the SVM over the CRF, and improvements in the LSTM over the SVM. We have also evaluated the results on ambiguous as well as ambiguous tokens. Figure 3 illustrates the various device performance for these tokens types. The results were higher for unclear tokens and reached 95.02% accuracy.

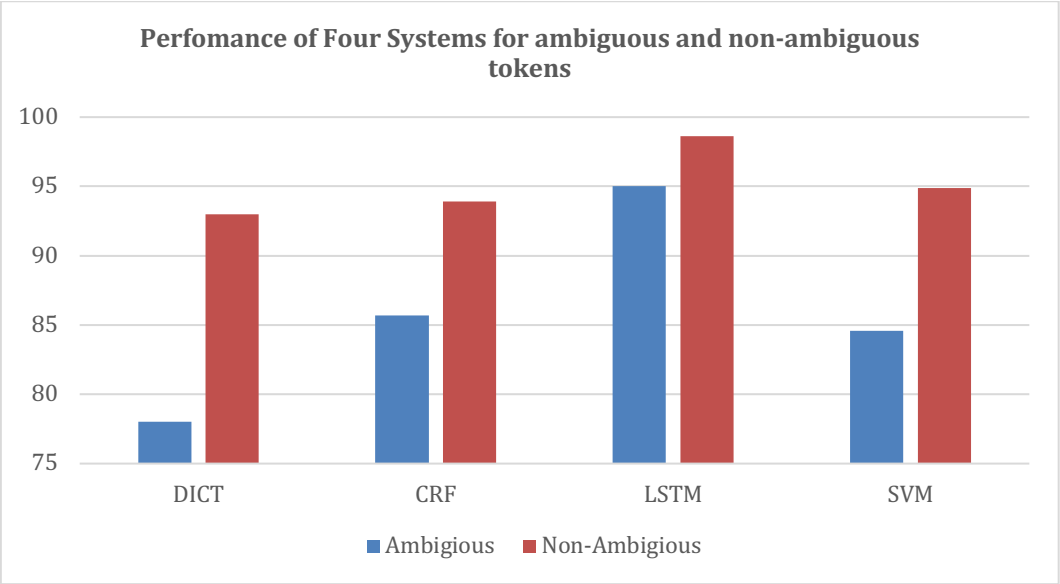

Fig. 3 Performance of Dictionary (DICT), CRF-based system (CRF), Long Short-Term Memory (LSTM) and Support Vector Machine (SVM) for ambiguous and non-ambiguous token.

This is accompanied by the results of 85.7% for CRF, SVM 85.58%, and the dictionary method of 78%. In the case of all the systems for an unambiguous token, i.e. 98.64%, LSTM reaches the greatest degree of uncertainty, but the result is comparable for SVM (94.89%), dictionary method (93.11%), CRF (93.91%) token.

5. Conclusion & Futuristic Scope

The problem in recognizing or detecting the language present in a given document or statement is referred to as language recognition or detection The Corpus of English and Roman Urdu is collected from social media websites. The Corpus of English and Roman Urdu is collected from social media websites. The text for training shall be submitted to a bi-direction LSTM accordingly to verify if the text is in English language or Urdu language. The results of word recognition for bidirectional word-level LSTM from Roman Urdu and English showed improved results.

The model proposed in the research work can be implemented for other languages and the same can be applied to different types of multilingual text to identify the language encapsulated within

the statement. The research work can be used with a large sample size from different dialects of the Roman-Urdu language to understand the performance of the model across horizontals and verticals.

Acknowledgment

The author extends the greetings and thanks to Dr. Mudasir M. Kirmani, SKUAST Kashmir Assistant Professor of Computer Science, Dr. Riyaz Ahmad Kumar, Assistant Professor (English), DDE, University of Kashmir, Mrs. Mudasir Hassan, Research Scholar, CCAS, the University of Kashmir without their suggestions and supports this work is impossible to complete. This work is not possible without the experts from English and Urdu Languages and experts from Computer Science.

Conflict of Interest

We have no conflicts of interest with this publication and there has been no significant financial support for this work that could have influenced its outcome. As Corresponding Author, I confirm that the manuscript has been read and approved before submission by all the above-named authors. We declare that this manuscript is original, has not been published before, and is not currently being considered to be published anywhere.

References

[1] A. Kralisch, and T. Mandl, “Barriers to information access across languages on the internet: Network and language effects”, In *System Sciences*, 2006.

[2] A. Xafopoulos, C. Kotropoulos, G. Almpanidis, and I. Pitas, “Language identification in web documents using discrete hmms”, In *Pattern recognition*, 37(3):583–594. (2004).

[3] B. Hughes, T. Baldwin, S. Bird, J. Nicholson, & A. MacKinlay, “*Reconsidering language identification for written language resources*”, In *5th International Conference on Language Resources and Evaluation*, LREC 2006, Genoa, Italy. Pp 485-488.

[4] B. King, and S. Abney, “Labelling the languages of words in mixed-language documents using weakly supervised methods”, In *Proceedings of the 2013 Conference of Human Language Technologies*. Association for Computational Linguistics. Atlanta, Georgia. pages 1110–1119, 2013.

[5] C. Lignos, and M. Marcus, “Toward web-scale analysis of codeswitching”, In *87th Annual Meeting of the Linguistic Society of America*. (2013).

[6] C. R. Voss, S. Tratz, J. Laoudi, and D. M. Briesch, (2014), “Finding romanized arabic dialect in code-mixed tweets”, In LREC, pages 2249–2253.

[7] D. Nguyen, and A. S. Dogruoz, “Word Level Language Identification in Online Multilingual Communication”, In *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* pages 857–862, Seattle, Washington, USA, Association for Computational Linguistics. 18-21 October, 2013.

[8] D. Elworthy, “Language identification with confidence limits” In *Springer*. 1999.

[9] E. Baykan, M. Henzinger, and I. Weber, (2008), “Web page language identification based on urls”, In *Proceedings of the VLDB Endowment*, 1(1):176–187.

[10] E. Tromp and M. Pechenizkiy, “Graph-based n-gram language identification on short texts”, In *Proc. 20th Machine Learning conference of Belgium and The Netherlands*, pages 27–34. (2011).

[11] G. Chittaranjan, Y. Vyas, K. Bali, and M. Choudhury, “Word-level language identification using crf: Code-switching shared task report of msr india system”, In *Proceedings of The First Workshop on Computational Approaches to Code Switching,* pages 73–79. (2014).

[12] G. Grefenstette, “Comparing two language identification schemes”, In JADT 1995, 3rd International conference on Statistical Analysis of Textual Data, Rome, Dec 11–13, 1995.

[13] H. Yamaguchi, and K. Tanaka-Ishii, “Text segmentation by language using minimum description length”, In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Long Papers-Volume 1, pages 969– 978. 2012.

[14] J. M. Prager, “Linguini: Language identification for multilingual documents” In *Journal of Management Information Systems*, pp 71–101. 16 Mar, 1999.

[15] K. N. Murthy, and G. B. Kumar, “Language identification from small text samples” In *Journal of Quantitative Linguistics*, 13(01):57–80. (2006).

[16] L. Grothe, E. W. De Luca and A. Nurnberger, (2008). A comparative study on language identification methods. In LREC. Citeseer.

[18] M. Damashek, “Gauging similarity with n-grams: Language-independent categorization of text” In Science, pp267-843. 1995.

[19] M. Diab, P. Fung, M. Ghoneim, J. Hirschberg, and T. Solorio, “Proceedings of the second workshop on computational approaches to code switching” In *Proceedings of the Second Workshop on Computational Approaches to Code Switching.* (2016).

[20] M. Goldszmidt, M. Najork, and S. Paparizos, “Boot-strapping language identifiers for short colloquial postings”, In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 95–111. Springer. (2013).

[20] M. Lui, and T. Baldwin, “langid. py: An off-the-shelf language identification tool”, In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30. Association for Computational Linguistics. (2012).

[21] M. Padro, and L. Padro, “Comparing methods for language identification”, Procesamiento del lenguaje natural, 33. (2004).

[22] P. McNamee, “Language identification: a solved problem suitable for undergraduate instruction” In *Journal of Computing Sciences in Colleges*, pp 94–101. 20 Mar 2005.

[23] R. Sequiera, M. Choudhury, P. Gupta, P. Rosso, S. Kumar, S. Banerjee, S. K. Naskar, S. Bandyopadhyay, G. Chittaranjan, A. Das, et al, “Overview of fire-2015 shared task on mixed script information retrieval”, In *FIRE Workshops*, volume 1587, pages 19–25. 2015.

[24] S. Bergsma, P. McNamee, M. Bagdouri, C. Fink, and T. Wilson, “Language identification for creating language-specific twitter collections”, In *Proceedings of the second workshop on language in social media*, pages 65–74. Association for Computational Linguistics. (2012).

[25] S. Carter, W. Weerkamp, and M. Tsagkias, “Microblog language identification: Overcoming the limitations of short, unedited and idiomatic text” Language Resources and Evaluation, 47(1):195–215. (2013).

[26] S. H. Kumhar, M.M. Kirmani, J. Sheetlani and M. Hassan, “Word Embedding Generation Methods and Tools: A Critical Review” in *International Journal of Innovative Research in Computer and Communication Engineering,* e-ISSN: 2320-9801, p-ISSN: 2320-9798, Volume 8, Issue 10, October 2020. DOI: 10.15680/IJIRCCE.2020.0810002

[27] S. Johnson, “Solving the problem of language recognition” In *Technical Report.* School of Computer Studies, University of Leeds. 1993.

[28] S. Rijhwani, R. Sequiera, M. Choudhury, K. Bali, and C. S. Maddila, “Estimating code-switching on twitter with a novel generalized word-level language detection technique”, In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), volume 1, pages 1971– 1982. (2017).

[29] T. Baldwin, and M. Lui, “Language identification: The long and the short of the matter in Human Language Technologies”, In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 229–237.2010.

[30] T. Dunning, “Statistical identification of language” In *Computing Research Laboratory*, New Mexico State University. 1994.

[31] T. Solorio, and Y. Liu, “Learning to predict code-switching points” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. pages 973–981. 2008a.

[32] W. B. Cavnar, and J. M. Trenkle, “N-gram-based text categorization. In *Proceedings of SDAIR-94, Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175. 1994.

[33] 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, pages 50–59. (2016).