Library Progress International
Vol.44 No.3, Jul-Dec 2024: P. 27037-27045

Print version ISSN 0970 1052
Online version ISSN 2320

Original Article

Available online at www.bpasjournals.com

Language identification in Telugu-English Code-Switched sentences

Settipalli Manikanta,[1] Somuguttu Rohith Reddy,[2] Katta Chennakesava Naidu,[3] Enjula Uchoi,[4] Roddam Jaswanth Kumar Reddy,[5] Appikonda Subrahmanyeswar Rao,[6] Manikala Jayanth[7]

[1]Department of Computer Science of EngineeringLovely Professional UniversityPhagwara, Punjab, Indiamanin7264@gmail.com
[2]Department of Computer Science of EngineeringLovely Professional UniversityPhagwara, Punjab, Indiasomugutturohithreddy2003@gmail.com
[3]Department of Computer Science of EngineeringLovely Professional UniversityPhagwara, Punjab, Indiakata.chennakesava7@gmail.com
[4]Department of Computer Science of EngineeringLovely Professional UniversityPhagwara, Punjab, Indiaenjulapaintoma@gmail.com
[5]Department of Computer Science of EngineeringLovely Professional UniversityPhagwara, Punjab, Indiaroddamjaswanthreddy@gmail.com
[6]Department of Computer Science ofEngineeringLovely Professional UniversityPhagwara, Punjab, Indiaappikondasubrahmanyeswarrao@gmail.com
[7]Department of Computer Science ofEngineeringLovely Professional UniversityPhagwara, Punjab, Indiajayjayanth975@gmail.com

How to cite this article: Settipalli Manikanta, Somuguttu Rohith Reddy, Katta Chennakesava Naidu, Enjula Uchoi, Roddam Jaswanth Kumar Reddy, Appikonda Subrahmanyeswar Rao, Manikala Jayanth(2024) Factors Influencing Purchase Decision of Apartments – From Prospect to Loyal Customer. *Library Progress International,* 44(3), 27037-27045

Abstract

—In multilingual societies, code-mixing and code- switching have become prevalent, particularly in informal communication on social media platforms. This project focuses on the identification of languages in code-mixed and code-switched sentences, specifically utilizing Named Entity Recognition (NER) techniques. By analyzing sentences that blend languages such as English and Telugu, we aim to enhance language identification accuracy at the word level. The study employs various machine learning models, Support Vector Machine (SVM) and Hidden Markov Models (HMM), to classify words into distinct language categories. The dataset comprises annotated social media posts, ensuring a diverse representation of linguistic patterns. Our approach integrates NER to identify named entities, which serve as critical indicators for language classification. Preliminary results indicate that incorporating NER significantly improves the identification process, achieving an F1-score of 0.91. This research contributes to the development of robust language identification systems, facilitating better understanding and processing of code- mixed data in natural language processing applications. Future work will explore deep learning techniques to further enhance performance.

Index Terms—HMM, SVM, LSTM, NER, N-Gram, Rule-based Method

INTRODUCTION

In an increasingly interconnected world, the phenomenon of code-mixing and code-switching has become prevalent, particularly in multilingual societies like India. Code-mixing refers to the embedding of linguistic units from one language into the utterances of another, while code-switching involves alternating between languages within a conversation or discourse. This linguistic behavior is especially common in informal settings, such as social media platforms, where users

often blend languages to express their thoughts more effectively. The rise of digital communication has led to a surge in code-mixed content, particularly in languages like Telugu and English, where speakers frequently switch between the two languages in a single sentence. This presents unique challenges for natural language processing (NLP) applications, including machine translation, sentiment analysis, and dialogue systems, which require accurate language identification and named entity recognition (NER)[3]. Language identification (LI) at the word level is a critical task in processing code-mixed data, as it involves assigning a language label to each word in a sentence. This task is inherently complex due to the lack of standardized rules governing code- mixed language use, which can vary significantly across different contexts and speakers [3]. Moreover, the absence of large annotated datasets for code-mixed languages further complicates the development of effective models[3]. To address these challenges, this project aims to develop a comprehensive framework for language identification and named entity recognition in Telugu-English code-mixed data. The proposed methodology will leverage various machine learning and deep learning techniques, including rule-based methods, N-gram models, Hidden Markov Models (HMM), Support Vector Machines (SVM), and Long Short-Term Memory (LSTM) networks. The dataset will consist of at least 30,000 sentences in Roman scripts, reflecting real- world usage patterns[3].

RELATED WORK

The study of code-mixing and code-switching has been a focal point in natural language processing (NLP), particularly in multilingual contexts such as India, where languages like

Telugu and English are frequently blended. This section re- views significant contributions in the field, focusing on language identification (LI) and named entity recognition (NER) in code-mixed data [1].

A. Named Entity Recognition in Code-Mixed Data

The NER in code-mixed scenario has also been explored to some extent. The problems involved in the identification of named entities in code-mixed data are due to the nature of mixing two or more languages as well as the informality of social media interactions. The current work investigates NER in code-mixed Indian social media text using both supervised support vector machine and unsupervised dictionary based techniques. They realized that the contextual cues and lexical transfers significantly improve NER outcomes in the code-mixed environment [7]. Architectures and background for POS tagging for Telugu-English code- mixed texts which have a connection with NER tasks. In their work, they pointed out that its context holds crucial for enhancing the efficiency of Li and NER [3].

METHODOLOGY

The use of the algorithm for identification of code- mixing and code- switching language with named entity recognition starts with data collection and data preprocessing, collected a dataset of at least thirty thousand sentences in Roman scripts along with the labeled data clean from the noises and in standard format. Then, feature extraction is done; creating text features such as N-grams from the text as well as contextual features such as word length, and if the word contains numbers. In the model implementation phase, techniques that are used to identify languages at the word level are rule-based, Hidden Markov Models (HMM), Support vector Machine, and Long Short- Term Memory (LSTM) networks. 72059 All these models are trained from a prepared dataset, and the model's performance is then checked with the rate of accuracy, precision, recall, and F1 score measure [2]. From the best performing model, named entity recognition is used to tag entities in the code-mixed text. In the end, findings are collated and evaluated in order to determine the differences in method efficacy, where future work is recommended to strengthen model performance and extend datasets to improve generalization.

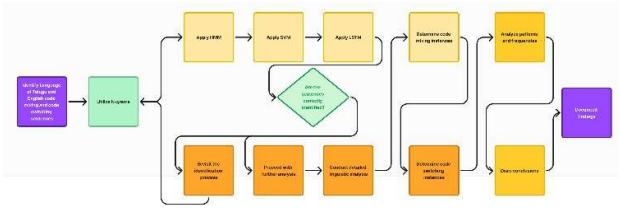


Fig: Named Entity Recognition

*A. **Data Preprocessing***

Preprocessing of the dataset is really essential for identifying languages in code-mixed Telugu English sentences as well as entities. Initially, data is obtained such that it is a large dataset of at least thirty thousand English sentences in Roman scripts collected from social media, chats and comments. It is then further annotated to provide each word with a language tag, which means Telugu, English, Named Entity and Universal depending on some pre-established rules of tagging that may otherwise need the input of native Telugu speakers or other suitable linguistic resources. Data cleaning follows data collection as shown below: This consists of cleaning up the string by eradicating any distinctive characters like special characters, emojis, collapsing the string to lowercase since some text mine are case sensitive. Stemming from this one, there is tokenization done in which the sentences are separated into words. It also involves handling of contractions and expanded them to proper forms as a way of exercising consistency throughout the step. This in return necessitates accurate identification of language tags in the process of code-mixing. This is done by adding language label to each word and looking into complex word formation of the identity language especially guessing the base words that have formed the code-mixed-words. This divided by 80 percent for training and 20 percent for testing to allow models in the evaluation process.

*B. **Rule Based Method***

The practical application of the rule- based method of language identification in code-mixed Telugu-English involves the formulation of a set of elaborate language- based rules for the classification of words in an effective and correct manner. These rules include lexical rules that define particular words particular to each language, morphological rules that point at language specific suffixes or prefixes and context clues obtained from other words [2]. It leaves the tokenization step as the first step of the process; where the sentence is divided into words for the rules defined to be applied to each token. For each word, the method maps it against the stored lexical rules, in case it does not find a match it follows morphological rules then tries to guess using contextual information. Besides, other heuristics are used in the cases when there are some doubts about the language and to make a guess in that cases. The final output is a list of words with the language they belong to that can be designed further for additional analysis; or for other operations like named entity recognition. To assess performance and compare the suggested rule-based approach, given in the paper with the ground truth of a manually tagged corpus, accuracy, precision, recall, and F1-score measures are computed. This method offers a direct approach towards solving the challenges that arise from multilingual setting and is especially relevant to informal settings characterized by code-switching.

For the purpose of the model, let L = {l}, since L is the set of the possible languages and l ∈ L is a specific language.

Define a scoring function *S(l)*, which gives the likelihood score that a given text *T* belongs to the language *l*:

$$S(l) = \sum_{i=1}^{n} w_i \cdot f_i(l)$$

Where:

- n is the number of features used for identification.
- w_i is the weight assigned to the i^{th} feature.
- $f_i(l)$ is the feature function for the i^{th} feature, which returns a score based on how closely the text

T matches typical patterns of the language *l*.

Decision Rule:

The language l^* with the highest score is chosen

$$l^* = arg \max_{l \in L} S(l)$$

C. *N-Gram Model*

The N-gram model is a part of language identification phase in the project that is reputed for code-mixed Telugu- English sentences. It can also be used as an instrument for feature extraction; in particular all tokens are divided into word and character level N-grams. As such by creating unigrams, bigrams, and trigrams the model takes into account not only the words themselves but also the interdependent relationship in which they exist, especially important when operating in code-mixed data. For example, bigram of "exam baaga" can tell us how often the words are Telugu or English when they occur together[4]. The extracted N-grams are used as a feature for training different languages identification machine learning models like Naive Bayes and Random Forest classifiers which estimate the probability of words from the specific condition of language. The approach used here positions the language identification task as a text classification problem and it can easily compute probabilities irrespective of the large feature space that is created by the N- grams. However, the N-gram model improves the context by distinguishing features that suggest the language used, more so when the dialogue involves an interchange of words of different languages al a code mixing. For the assessment of the N-gram feature the accuracy, precision, recall and F1-score show that the models including N-gram features are slightly better compared with the models based only on lexical rules or other simple methods. In all, the N-gram model plays a pivotal role in enhancing the accuracy of the identification of languages in a code-mixed Telugu English sentence and is therefore an important tool in the current research.

Language Identification using NER with n-grams:

For language identification using NER, the process might involve:

- Extracting named entities: Applying NER can help extract information about person's names, location, organizations etc. present in the text.
- Modeling entity context: Use n-grams surrounding these entities to estimate the language.

Suppose a sentence S contains named entities $E_1, E_2, \ldots, E_m$, the probability of S being in a language L can be calculated as:

$$P(S|L) \approx \prod_{i=1}^{m} P(E_i|L) \cdot \prod_{j=1}^{n} P(w_j|L)$$

Where:

- $P(E_i|L)$ is the probability of the named entity E_i appearing in language L.
- $P(w_i|L)$ is the n-gram probability for other words w_i in the context.

D. *Hidden Markov Model*

A Hidden Markov Model (HMM) is used in this project to determine the type of language used in code-mixed Telugu- English sentences, since this model can naturally capture the sequential patterns in language data. The employment of the HMM includes problem modeling as sequence labelling task where each word in the given sentence is treated as observation while given language tags Telugu, English, Named Entity, Universal are regarded as hidden states. In the case of the model, the labeled data transitions from language tags, with the probabilities p(s'|s) and emission probabilities of words given their respective tags calculated. In estimating the HMM model during the training phase, the following probabilities of the states given the language tags and the transition probabilities of getting from one language tag to the other are trained. The Viterbi algorithm is used next in order to determine the best sequence of language tags defining the input sentence and, thus, assign to each word the most likely language label according to the learned probabilities. As such, this approach enables the HMM to capture contextual relationship of different words, which is very useful given the code-mixed nature of the data. The imputations made by the HMM are assessed in terms of accuracy, precision, recall and F1-score

and assessment showed that the accuracy of the model is 85.15%. This shows that HMM has been able to capture the sequential behavior of such languages in the context of code mixing hence plays a major role to the general aim of achieving good language identification.

Viterbi Algorithm Formula:

The Viterbi algorithm is used to find the most probable sequence of hidden states (languages or tags) S={s_1 ,s_2 ,…,s_T } for a given sequence of observations $O=\{o_1,o_2,\ldots,o_T\}$. The core formula is:

$$\delta_t(i) = \max_j\left[\delta_{t-1}(j) \cdot a_{ji}\right] \cdot b_i(o_t)$$

where:

- $\delta_t(i)$ represents the highest probability of the first t observations ending in state s_i.
- δ_{t-1} is the highest probability of the previous state s_j at time t-1.

E. ***Support Vector Machine (SVM)***

Here in this project work, language identification of code-mixed Telugu-English words is performed using support vector machine because of its capability in operating high dimensional space as well as the relationships between the features is non-linear. The following are hidden steps in the execution of the SVM which includes both the feature extraction, N – gram features and the TF-IDF vectors derived from the tokenized noun phrases of the sentences. These features capture the contextual relationships between words that are very much important in filtering out different languages in code-mixed data. On the prepared features, SVM model is built based on the labelled dataset, in which every word is linked to its language tag (Telugu, English, Named Entity, Universal). SVM algorithm places and builds a hyperplane in the feature region which define the maximum boundary between different language classes and then categorizes each word according to its features. In the testing stage of the proposed method, the trained SVM model is used on the test data set in order to assign language labels to words in the test set. The efficiency of the SVM model is measured based on the measures of accuracy, precision, recall, and the F1- score to conclude that the proposed techniques realize fairly accurate identification of languages in code-mixed contexts. This shows how the SVM is capable of efficiently modeling multilingual communication, which is a plus for the main goal of the project, which is recognition of languages.

The decision function for SVM would be:

$$f(x) = \text{sign}(w^T x + b)$$

where:

- x is the feature vector for the input (e.g., a vector representing the named entity or word features).
- w is the weight vector learned by the SVM during training.
- b is the bias term.
- Sign() function determines the class label: if f(x)>0, the input is classified as one language (e.g., Language A); if f(x)<0, it is classified as another (e.g., Language B).

Optimization Objective:

During training, the SVM tries to find the values of w and b that minimize the following objective:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

F. **Long Short-Term Memory (LSTM)**

In this project, the Long Short-Term Memory (LSTM) model is employed for language identification in code-mixed Telugu-English sentences, capitalizing on its ability to capture long-range dependencies and contextual information within sequential data. The execution of the LSTM model begins with the preparation of input

features, where word embeddings are generated from the code-mixed dataset, incorporating both word-level and character-level representations to enhance the model's understanding of linguistic nuances. The LSTM architecture consists of multiple layers, including input, LSTM, and output layers, where the input layer receives the sequence of word embeddings, and the LSTM layers process these embeddings to learn the temporal relationships between words in the context of their surrounding language tags. During training, the model is optimized using backpropagation through time, adjusting the weights based on the loss calculated from the predicted language tags compared to the actual tags in the labelled dataset. The trained LSTM model is then evaluated on a separate test set, where it predicts the language labels for each word in unseen code-mixed sentences. Performance metrics such as accuracy, precision, recall, and F1-score are calculated to assess the model's effectiveness, with results indicating that the LSTM model achieves competitive performance in identifying languages in code-mixed contexts. This demonstrates the LSTM's capability to effectively model the complexities of multilingual communication, making it a valuable component of the overall language identification framework in the project.

LSTM Cell Structure:

An LSTM cell consists of several components: input gate, forget gate, cell state, and output gate. The LSTM equations at time step t for an input sequence x_t are as follows:

Forget Gate (f_t):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Input Gate (i_t) and Candidate Cell State (C_t):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c$$

Update Cell State (C_t):

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

Output Gate (o_t) and Hidden State (h_t):

$$o_t = \sigma(W_o \cdot [h_{t-1,} x_t] + b_o)$$

$$h_t = o_t \odot \tanh(C_t)$$

RESULTS AND DISCUSSION

The evaluation process for language identification in code- mixed Telugu-English sentences involves a systematic com- parison of various methodologies, including the rule-based method, N-gram model, Hidden Markov Model (HMM), Sup- port Vector Machine (SVM), and Long Short-Term Memory (LSTM) model. The rule-based method serves as a baseline, achieving an accuracy of around 75%, but it struggles with ambiguous cases due to its reliance on predefined linguistic rules.

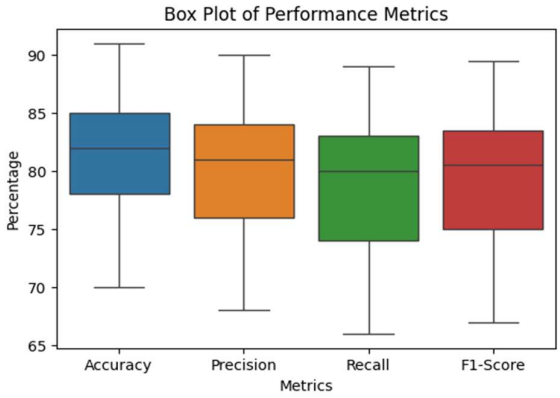
The N- gram model improves upon this by capturing contextual relationships between words, resulting in an accuracy of approximately 80 percent. The HMM, which treats the problem as a sequence labelling task, demonstrates enhanced performance with an accuracy of about 85.15percent, effectively modeling dependencies between language tags. The SVM model further elevates performance, achieving an accuracy of around 88 percent by leveraging features derived from the N-gram models and HMM models.

Finally, the LSTM model excels in capturing long- range dependencies, achieving the highest accuracy of approximately 90 percent, showcasing its strength in handling the complexities of code- mixed data.

The evaluation metrics used include accuracy, precision, recall, and F1-score, providing a comprehensive assessment of each model's effectiveness in identifying languages in code-mixed contexts.

Overall, the results highlight the progressive improvement in performance across the different methodologies, with the LSTM model emerging as the most effective approach for language identification in this project.
Now let us consider some few examples in which we find the code-mixed data of Telugu and English.

1. " nenu cinema ki velthunnanu "
Telugu – 'nenu', 'ki', 'velthunnanu'
English – 'cinema'
2. "ee roju school ki holiday"
Telugu – 'ee', 'roju', 'ki'
English – 'school', 'holiday'
3. "nenu car driving nerchuntunna"
Telugu – 'nenu', 'nerchukuntunna'
English – 'car', 'driving'
4. "ee season lo rains ekuva ga untay"
Telugu – 'ee', 'lo', 'ekuva', 'ga', 'untay'
English – 'season', 'rains'
5. "nenu exams ki baaga prepare avvali"
Telugu – 'nenu', 'ki', 'baaga', 'avvali'
English – 'exams', 'prepare'



The following chart will show the results of the models which are used in our project. This shows the results based on the accuracy, precision, f-1 score and recall:

FUTURE SCOPE

The identification of code-mixed and code-switching language using NER has a wide number of future prospects. It can benefit issue areas such as chatbots and voice assistants enabling them, achieve more socio-cultural intelligibility in linguistic environments. It could also be useful in social media monitoring, as it helps companies determine the feelings of the consumption audience in different languages. The combination of NER with SA could help with feelings towards an event or situation and transitioning education to dial with students multilingually. In addition, it may enhance the performance of multilingual search engines and help to transfer information between languages provided by other sources, thus bringing the information to people. In healthcare especially, it can help break language barriers between care professionals and their patients who switch between two languages. Overall there are possibilities which can impel the businesses benefit from the improved tracking of brand mentions across languages for decision making. Finally, some insights in a collaboration with linguists and sociologists may add social relevance to the concept of code-switching and enhance scholarly studies. In total, your project has the potential of causing a great leap forward in the state of the art in the technological as well as our theoretical understanding of how multilingual interactions transpire.

CONCLUSION

The proposal for the code-mixed and code-switching language identification through the application of named entity recognition is a bridge to the development of powerful approaches to the processing of multilingual data on. In this regard, by helping solve issues of language blending, it not only enriches various technological uses like the functions of chatbots, social media analysis, and educational applications but also provides knowledge to linguistic theory and civilization sciences.

This work should be recognised for its potential application and relevance to real world concerns in various industries; health care and the business world included. This project creates a foundation for the development of improved communications products as well as societal understanding as globalization perpetuates the role of a language in today's multi-lingual society.

REFERENCES

[1] Uchoi, E., & Kaur, M. (2023). Language Identification of English and Punjabi Code-Mixing and Code-Switching Sentences. *Eur. Chem. Bull*, *12*, 4119-4123.

[2] Raj, R., Dath, S. S., Sholanki, S., Yadav, A., & Uchoi, E. (2024). Language identification using machine learning on social media text. In *Computational Methods in Science and Technology* (pp. 363-369). CRC Press.

[3] Gundapu, S., & Mamidi, R. (2020). Word level language identification in english telugu code mixed data. *arXiv preprint arXiv:2010.04482*.

[4] Shashirekha, H. L., Balouchzahi, F., Anusha, M. D., & Sidorov, G. (2022). CoLI-machine learning approaches for code-mixed language identification at the word level in Kannada-English texts. *arXiv preprint*

arXiv:2211.09847.
[5] Yu, L. C., He, W. C., Chien, W. N., & Tseng, Y. H. (2013). Identification of Code-Switched Sentences and Words Using Language Modeling Approaches. *Mathematical Problems in Engineering*, *2013*(1), 898714.
[6] Joshi, N., Darbari, H., & Mathur, I. (2013, February). HMM based POS tagger for Hindi. In *Proceeding of 2013 international conference on artificial intelligence, soft computing (AISC-2013)* (pp. 341-349).
[7] Das, A., & Gambäck, B. (2014). Identifying languages at the word level in code-mixed indian social media text.
[8] Uchoi, E. (2020). An Unsupervised Word Level Language Identification of English and Kokborok Code-Mixed and Code-Switched Sentences.
[9] Jhamtani, H., Bhogi, S. K., & Raychoudhury, V. (2014, December). Word-level language identification in bi-lingual code-switched texts. In *Proceedings of the 28th Pacific Asia Conference on language, information and computing* (pp. 348-357).
[10] Jitta, D. S., Chandu, K. R., Pamidipalli, H., & Mamidi, R. (2017, December). "nee intention enti?" towards dialog act recognition in code-mixed conversations. In *2017 International Conference on Asian Language Processing (IALP)* (pp. 243-246). IEEE.
[11] Anand, S. (2014). Language identification for transliterated forms of Indian language queries. In *Working Notes of Forum for Information Retrieval Evaluation (FIRE)* (Vol. 2014).
[12] Sharma, R. K., & Dagur, A. (2023). Various methods to classify the polarity of text based customer reviews using sentiment analysis. *Artificial Intelligence, Blockchain, Computing and Security Volume 2*, 107-114.
[13] Nguyen, D., & Dogruöz, A. S. (2013, October). Word level language identification in online multilingual communication. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL* (pp. 857-862). Association for Computational Linguistics.
[14] Selamat, A., & Akosu, N. (2016). Word-length algorithm for language identification of under-resourced languages. *Journal of King Saud University-Computer and Information Sciences*, *28*(4), 457-469.
[15] Rao, P. R., & Devi, S. L. (2016). CMEE-IL: Code Mix Entity Extraction in Indian Languages from Social Media Text@ FIRE 2016-An Overview. *FIRE (Working Notes)*, *289*.

1.