Entity Extraction of Hindi-English and Tamil-English Code-Mixed Social Media Text

G. Remmiya Devi(✉), P. V. Veena, M. Anand Kumar, and K. P. Soman

Centre for Computational Engineering and Networking (CEN),
Amrita School of Engineering,
Amrita Vishwa Vidyapeetham, Coimbatore, India
remmiyanair@gmail.com, veenakrt27@gmail.com, m_anandkumar@cb.amrita.edu,
kp_soman@amrita.edu

Abstract. Social media play an important role in today's society. Social media is the platform for people to express their opinion about various aspects using natural language. The social media text generally contains code-mixed content. The use of code-mixed data is popular in them because the users tend to mix multiple languages in their conversation instead of using their native script as unicode characters. Entity extraction, the task of extracting useful entities like Person, Location and Organization, is an important primary task in social media text analytics. Extracting entities from code-mixed social media text is a difficult task. Three different methodologies are proposed in this paper for extracting entities from Hindi-English and Tamil-English code-mixed data. This work is submitted to the shared task on Code-Mix Entity Extraction for Indian Languages (CMEE-IL) at the Forum for Information Retrieval Evaluation (FIRE) 2016. The proposed systems include approaches based on the embedding models and feature-based model. BIO-tag formatting is done as a pre-processing step. Extraction of trigram embedding is performed during feature extraction. The development of the system is carried out using Support Vector Machine-based machine learning classifier. For the CMEE-IL task, we secured second position for Tamil-English data and third for Hindi-English. Additionally, evaluation of primary entities and their accuracies were analyzed in detail for further improvement of the system.

Keywords: Social media text · Entity extraction · Code-mixed text
Word embedding · Trigram embedding features
Support Vector Machine

1 Introduction

Entity extraction has been a primary task in most of the Natural Language Processing (NLP) applications and it is defined as a task of extracting the named entities from any text. Entities of major categories are Person, Location, and

© Springer International Publishing AG 2018
P. Majumder et al. (Eds.): FIRE 2016 Workshop, LNCS 10478, pp. 206–218, 2018.
https://doi.org/10.1007/978-3-319-73606-8_16

Organization. The other entities are count, artifact, year, entertainment etc. The social media text, although unstructured contains informative content like the review about a product or a movie or opinion about a person or organization. Extracting entities from such informative content is the most challenging task. Moreover, the posts or comments in social media generally contain text in conversational nature which is in code-mixed format. The users also tend to type the text of their native language as Roman script instead of unicode characters.

An example for code-mixed Hindi-English text is given below:

chalo pehli baar **corporate** *mei bhi* **rainy day** *hoga.*

In the above example, English words are in bold letters and Hindi words are in italics. It can be seen that the Hindi words are in Roman script. Communication in social networking platforms like Facebook and Twitter are usually in code-mixed language.

This work is an extended version of the work submitted to shared task of CMEE-IL in FIRE 2016 [1]. In this shared task, we deal with Hindi-English and Tamil-English code-mixed dataset and the objective is to develop a system to extract entities from the content in code-mixed language.

Our proposed system for the task includes three approaches. The first approach is developed using word embedding features obtained from wang2vec model [2]. The second approach utilizes word embedding features from word2vec model [3]. The major difference between wang2vec and word2vec features lies in the inclusion of word order information in wang2vec. In the third approach, stylometric features were extracted from the training data. Extracted features from these three systems are used to develop three separate models using Support Vector Machine (SVM)-based classifier [4].

A brief description of previous related works is given in Sect. 2. Section 3 discusses on the code-mixed entity extraction task. The dataset statistics is explained in Sect. 4. Section 5 gives a brief overview on the word embedding models used in this paper. The methodology proposed is explained in Sect. 6. Discussion over experiments and results are explained in Sect. 7. The conclusions derived from the work is given in Sect. 8.

2 Related Work

In recent years, several researches were carried out in the field of text processing using code-mixed data. POS (Part-of-Speech) tagging was performed for code-mixed data of social media content in English-Hindi language [5]. A language identification task was carried out for Facebook data which is code-mixed between Bengali, English and Hindi [6]. A paper was published on thematic knowledge discovery for Facebook chat messages in English-Hindi using topic modeling [7]. Question classification system for code-mixed social media text in Bengali-English was developed using Bag of Words (BOWs) and Recurrent

Neural Networks (RNN) embeddings [8]. A hybrid approach of machine learning with rule based system was proposed for entity extraction in code-mixed English-Hindi and English-Tamil text [9].

Several works and shared tasks have been carried out in the field of entity extraction in social media platforms and few of them related to Indian Languages are enlisted here. In the Named Entity Recognition (NER) Task by Forum for Information Retrieval (FIRE)-2014, entity extraction for Indian languages like Hindi, Tamil and Malayalam was performed using rich features like context words, POS tags, root word, length and position [10]. FIRE 2015 entity extraction task focused on extracting named entities from social media text containing English, Malayalam, Tamil and Hindi content [11]. This work on entity extraction was implemented using SVM-based classifier in [12]. A Conditional Random Field (CRF) based approach was also implemented for the ESM-IL task of FIRE 2015 entity extraction task [13]. Another CRF-based system was proposed to perform named entity recognition for Indian Languages for Twitter based social media text [14]. With the popularity of code-mixed language in social media, a task on entity extraction from code-mixed social media data for Indian Languages (CMEE-IL) was conducted by AU-KBC Research Centre in FIRE 2016 [15]. In the CMEE-IL shared task, entity extraction for code-mixed data was implemented using neural networks [16]. A context-based character embedding was also implemented for the same entity extraction task [17]. International conferences for the shared task on entity extraction are listed below. An overview of Named Entity rEcognition and Linking in Italian Tweets for the task (NEEL-IT) was organized at EVALITA 2016 [18]. The Workshop on Noisy User-generated Text (WNUT16) conducted a Named Entity Recognition shared task for Twitter data in English language [19].

3 Code-Mixed Entity Extraction Task

The shared task on entity extraction of code-mixed languages was conducted by Computational Linguistics Research Group (CLRG), AU-KBC Research Centre, Chennai in FIRE 2016. The data for the task was collected from Twitter and other few microblogs. Hindi-English and Tamil-English code-mixed dataset was provided by the task organizers. Nine teams participated for entity extraction task in Hindi-English and five teams for Tamil-English entity extraction task. An example of Hindi-English and Tamil-English code-mixed tweet is shown in Table 1. The aim of the task is to extract named entities from the test data provided by organizers. Named entities in the dataset include Person, Location, Count, Organization, Year and so on. The number of users in social media platforms using code-mixed languages have increased predominantly today. Hence this task involving code-mixed language holds a significant relevance. The task evaluation was based on overall Precision, Recall, and F-measure. Since the individual entity-wise accuracy was not provided by the organizers, the participants could not figure out for which entity, the accuracy goes wrong and the reason why recall is less although precision value is more.

4 Dataset Description

Hindi-English and Tamil-English code-mixed text were used for the shared task. The three fields of the training dataset are Tweet ID, User ID, and the tweets. Each training file has a corresponding annotation file. The annotation file contains 6 fields namely Tweet ID, User ID, Length, Index, Entity tag and the entity chunks in the train data. Unlike Hindi-English dataset, Tamil-English dataset is not in complete code-mixed language, instead, some of the data are in pure Tamil script. To train the word embedding model, additional code-mixed text were used for both datasets. Data from the POS tagging task by International Conference on Natural Language Processing (ICON) 2015 [5], Mixed Script Information Retrieval 2016 (MSIR) [20], and some twitter data were the sources for the additional dataset for Hindi-English. The additional dataset for Tamil-English code-mixed language is from Sentiment Analysis in Indian Languages (SAIL-2015) [21,22]. The total number of tweets, average tokens per tweet and the number of entity chunks present in the training and testing data is tabulated in Table 2. On considering Hindi-English dataset, the number of tweets in test data is high compared to train data. Even though the number of tweets in Hindi-English train data is less than Tamil-English, the count of entity chunks is high in the Hindi-English dataset. The average tokens per tweet is high for Hindi-English than Tamil-English. This is due to the fact that the words in the Tamil language is generally agglutinative in nature.

The size of the additional dataset collected for Hindi-English and for Tamil-English is tabulated in Table 3. Since the time to complete the task was limited, we could not collect a surplus amount of Tamil-English code-mixed text.

Table 1. Example code-mixed text

Language	Code-mixed text
Hindi-English	Her reply pe muje smile Ati hai and I've the worst smile ever
Tamil-English	Intha padam parthe piragu than ennoda romba pudicha actor aanaru

Table 2. Number of tweets, average tokens per tweet and the number of entity chunks present in train and test data

	Train data			Test data		
	Tweet count	Avg tokens per tweet	No of entity chunks	Tweet count	Avg tokens per tweet	No of entity chunks
Hindi-English	2700	16.76	2413	7429	16.49	-
Tamil-English	3200	11.94	1624	1376	12.11	-

The Table 4 lists the six major entities present in the training data and their respective count. It can be observed from the table that one of the major entity, ENTERTAINMENT is covered more in Hindi-English code-mixed text.

Table 3. Size of additional utterances used

	Dataset size
Hindi-English	20617
Tamil-English	1625

Table 4. Major entity chunks and their count in the train dataset

Entities	Hindi-English	Tamil-English
Person	712	661
Location	194	188
Organization	109	68
Entertainment	810	260
Year	143	54
Count	132	94

5 Word Embedding Models

Word embedding is the vector representation of words. The purpose of word embedding is to map a word from a higher dimension to lower dimension through vector representation. Word2vec is a popular model used for retrieving word embedding features from text [3]. It includes two architectures. They are Continuous Bag of Words (CBOW) model and Skip-gram model. The improvised package of word2vec has been developed with additional features and named as wang2vec [2].

Sentences serve as input for training word embedding model. Vector representations acquired from the model of each word are based on the syntactic pattern in which the words lie in training sentences. The aim of the skip-gram is to predict the context word that has maximum likelihood given the neighboring words. So the aim is to maximize the value of X which is equated as in Eq. 1 where N is the total number of words, a is the window size and $p\left(y_{n+k}|y_n\right)$ is the output probability.

$$X = \frac{1}{N} \sum_{n=1}^{N} \sum_{-a \leq k \leq a} \log p\left(w_{n+k}|w_n\right) \tag{1}$$

To predict the context words $O \in w_{-a}, \ldots w_{-1}, w_1, \ldots w_a$ provided the center word w_0, the skip-gram utilizes a single output matrix $R \in |V| \times d$ where d is the embedding dimension.

The additional feature that improvised word2vec model to wang2vec model is the word order information. The sentences are fed to the neural network layer present in the word embedding model. The words are learned with respect to their position in the sentences. This enables the word embedding model to get trained in a better way. This, in turn, provides the system a better understanding of the syntactic pattern of each word. A set of $a \times 2$ output predictors $O_{-a}, \ldots O_{-1}, O_1, \ldots O_a$ of size $O \in |V| \times d$ [2].

The probability score of each word depicts the level of its relevance to the input word. The probability function used is softmax classifier and the corresponding equation is shown below in Eq. 2.

$$p(w_o \mid w_i) = \frac{e^{o_{w_i}(w_o)}}{\sum w \in V e^{o_{w_i}(w)}} \tag{2}$$

where, V is the vocabulary words and O_{w_i} corresponds to the $|V|$ dimensional vector.

6 Methodology for Entity Extraction

The preprocessing task is an essential step before analyzing and processing social media text. Initially, tokenization is performed on the given training data. The tokens obtained after the tokenization process is converted into conventional BIO format. This leads to BIO tag information for each word in the training data. Generally, BIO tag stands for Beginning, Inside, Outside tag. For example, consider the sentence, "*Sundar Pichai is the CEO of Google*". In general, the entity '*Sundar Pichai*' indicates PERSON and '*Google*' indicates ORGANIZATION. Since the word *Sundar Pichai* has two parts, it is tagged in BIO format

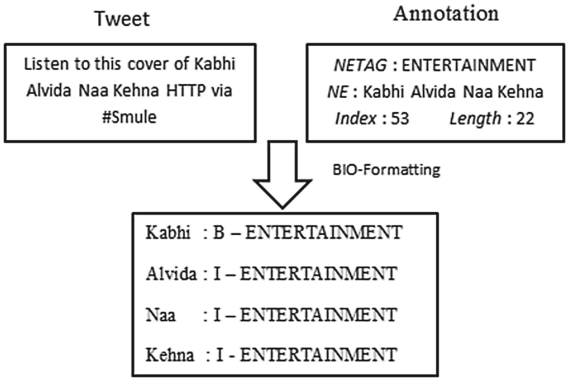

Fig. 1. Illustration of BIO-formatting

as beginning and inside. Outside tag is for the words that do not belong to any named entity. With the usage of BIO tag, words in the above example sentence, *Sundar* is labeled as B-PERSON, *Pichai* as I-PERSON and *Google* as ORGANIZATION. This BIO tag information is utilized in the three systems proposed in the paper. An illustration of BIO-formatting is given in Fig. 1.

The illustration of the proposed word embedding based models are shown in Fig. 2 and feature based method is shown in Fig. 3.

6.1 Entity Extraction with Wang2vec Embedding Features

The wang2vec model is an advanced version of the word2vec model with respect to the architecture of the models. The skip-gram model in the word2vec has become Structured Skip-gram model in wang2vec. The main improvement in this model is the fact that the word position information is taken into consideration. The wang2vec model is used to obtain the word vector features using structured skip-gram model. The vector size n is defined during the training of wang2vec model. Here, vector size n, for each word is set as 50. The word embedding features of each word in the given dataset are retrieved from the resultant vectors.

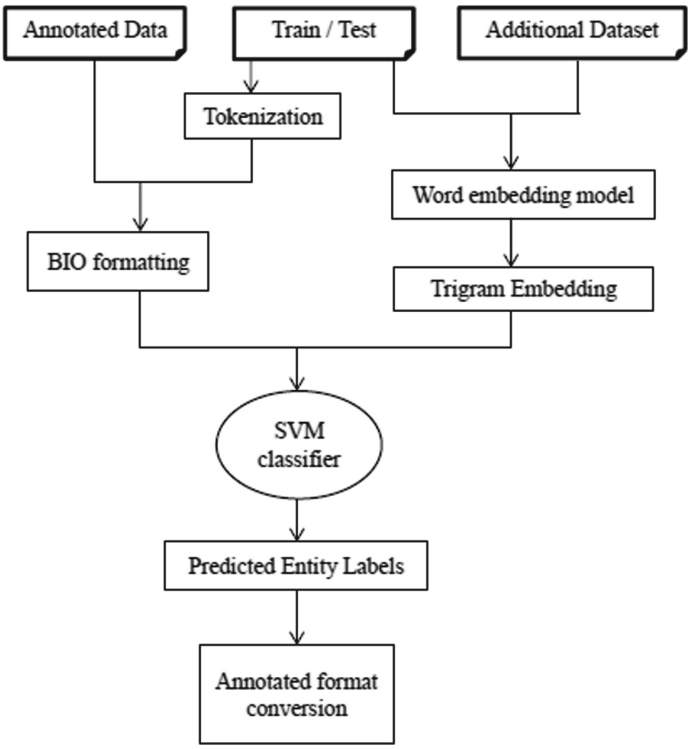

Fig. 2. Methodology of the entity extraction system using word embedding models

The neighboring context features from these vectors were retrieved and combined with the original embedding features. This, in turn, forms a feature set of size 150. Trigram embedding features are formed by appending original features to the context features. Finally, each word in the training data holds its corresponding BIO tag information and trigram embedding features. This serves as an input to train the SVM classifier. Hence an SVM model corresponding to this system is obtained. Wang2vec embedding features for test data were extracted in the same way. For testing, each word along with its trigram embedding feature set is given to the classifier.

6.2 Entity Extraction with Word2vec Embedding Features

The vector representation for each vocabulary word in the dataset can be retrieved using the well-known word embedding model called word2vec. Input for word2vec are sentences, as the extraction of word embedding solely depends on the syntactic pattern which is retrieved from the context words of each sentence. Word2vec uses the skip-gram model to obtain the vectors of each word in the training data. The entity extraction system is developed using these features. Similar to the system using wang2vec, this system also utilizes trigram embedding features of word2vec features.

Each word of the training data are merged with its feature set containing BIO tag information and the trigram embedding features. These features are integrated to form a feature set which is given for training the machine learning based classifier, SVM. Each word in the test data is merged with the trigram embedding features and is given for testing. The SVM classifier learns the different syntactic pattern in which each word lies from training data and performs recognition of entities from the dataset during testing of the system.

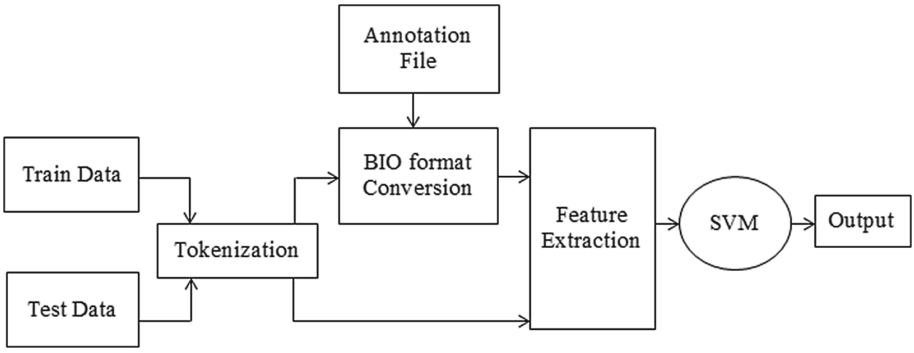

Fig. 3. Methodology of the entity extraction system using stylometric features

6.3 Entity Extraction with Stylometric Features

The third approach uses a conventional feature extraction method implemented using stylometric features. The stylometric feature set contains features like Prefix, Suffix, Length, Number, Lower case/Upper case and so on. The features used in our system are tabulated in Table 5. Each word in the training data is targeted and its corresponding stylometric feature set is extracted. The BIO tag information is joined with the conventional feature set to form the input for training SVM. For predicting the output tag, the stylometric features extracted are merged with the words in test data and given for testing.

The proposed methodology results with three SVM models for the three systems using wang2vec features, word2vec features and stylometric features.

Table 5. Stylometric features extracted from train and test data for feature based system

Features	Representation
Lower case	Lower case representation
Hyperlink	1 if hyperlink present
Hash (#), apostrophe (')	1 if word starts with #, ' symbol
Numbers, punctuation	Marks 1/0 if present/absent
Prefix-suffix	P3/P4: first 3/4 characters
	S3/S4: last 3/4 characters
Length and index	No of chars & position of word
4-digit numbers	1 if Token is a 4-digit number
First character upper case	1 if an upper case character
Full character upper case	1 if entirely an upper case word
Gazetteer features	Person, location, organization, entertainment

7 Experiments and Results

The procedure for wang2vec based embedding and word2vec based embedding models are similar. The difference between these two models is that wang2vec embedding features takes the word position into consideration using structured skip-gram model. The skip-gram model is utilized to retrieve word embedding features. The additional dataset is also taken along with the training data to train the two word embedding models. The vector size is set as 50. According to the user defined vector size, the trigram embedding feature set is extracted. The train data, after tokenization, is converted to BIO-formatted text. The words in the training data is merged with its trigram embedding vectors of size 150 and BIO-tag information serves as an input to the SVM classifier. Test data is also subjected to similar steps for retrieving trigram embedding feature set. After

tokenization, the trigram embedding features of length 150 is given to SVM for predicting label of test data.

The third approach uses stylometric features for entity extraction of the code-mixed data. From the tokenized train data, features listed in Table 5 are extracted. BIO tag information along with the stylometric feature set of these words are given to the classifier to train the system. The tokenized words are appended with its corresponding feature set and given to the classifier for testing.

The systems were trained with 10-fold cross validation. Table 6 shows the cross-validation results obtained for Hindi-English and Tamil-English dataset for wang2vec, word2vec, and feature based models. The accuracy for unknown tokens is better in wang2vec based embedding model.

From the results by CMEE-IL task organizers, for Tamil-English we have obtained the second place and for Hindi-English we ranked in the third place. Tables 7 and 8 tabulate the results of the top five teams for Hindi-English and Tamil-English data respectively. From the table, it can be inferred that the

Table 6. Cross validation results for Hindi-English and Tamil-English

	Hindi-English			Tamil-English		
	wang2vec	word2vec	Features	wang2vec	word2vec	Features
Known	92.99	91.10	94.26	97.27	97.38	97.49
Ambiguous known	83.09	78.32	86.56	83.81	83.84	85.97
Unknown	91.03	90.95	86.94	93.67	93.44	92.46
Overall accuracy	**92.47**	**91.03**	**92.37**	**96.15**	**96.25**	**95.98**

Table 7. CMEE-IL results by the task organizers for Hindi-English

Team	Run 1			Run 2			Run 3		
	P	R	F	P	R	F	P	R	F
Irshad-IIT-Hyd	80.92	59.00	68.24	-	-	-	-	-	-
Deepak-IIT-Patna	81.15	50.39	62.17	-	-	-	-	-	-
Amrita_CEN	**75.19**	**29.46**	**42.33**	**75.00**	**29.17**	**42.00**	**79.88**	**41.37**	**54.51**
NLP_CEN_Amrita	76.34	31.15	44.25	77.72	31.84	45.17	-	-	-
Rupal-BITS_Pilani	58.66	32.93	42.18	58.84	35.32	44.14	59.15	34.62	43.68

Table 8. CMEE-IL results by the task organizers for Tamil-English

Team	Run 1			Run 2			Run 3		
	P	R	F	P	R	F	P	R	F
Deepak-IIT-Patna	79.92	30.47	44.12	-	-	-	-	-	-
Amrita_CEN	**77.38**	**8.72**	**15.67**	**74.74**	**9.93**	**17.53**	**79.51**	**21.88**	**34.32**
NLP_CEN_Amrita	77.70	15.43	25.75	79.56	19.59	31.44	-	-	-
Rupal-BITS_Pilani	55.86	10.87	18.20	58.71	12.21	20.22	58.94	11.94	19.86
CEN@Amrita	47.62	13.42	20.94	-	-	-	-	-	-

overall accuracy of the Tamil-English is less than the Hindi-English. The reason for this might be the fact that the dataset of Tamil-English is not completely code-mixed and partially in pure Tamil script. In the results obtained by our system, we observed that the recall value is less compared to precision. This may be because less unlabeled data were collected for word embedding models.

Table 9. Precision, recall and F-measure obtained for major entities of Hindi-English code-mixed text

Entities	Precision	Recall	F-measure
PERSON	56.52	21.31	30.95
LOCATION	64.29	36.00	46.15
ORGANIZATION	50.00	23.81	32.26
ENTERTAINMENT	47.30	32.11	38.25
YEAR	60.00	20.00	30.00
COUNT	55.00	31.43	40.00

Table 10. Precision, recall and F-measure obtained for major entities of Tamil-English code-mixed text

Entities	Precision	Recall	F-measure
PERSON	50.00	35.24	41.34
LOCATION	77.77	26.92	40.00
ORGANIZATION	22.22	20.00	21.05
ENTERTAINMENT	35.29	24.00	28.57
YEAR	100.00	20.00	33.33
COUNT	55.88	55.88	55.88

After the shared task, we decided to perform detailed entity-wise error analysis. Since the gold-standard dataset is not available, we divided the training data further into train and test data. The analysis was performed on the feature based system only because compared to word embedding based systems, the system using traditional features has given better accuracy. Tables 9 and 10 shows the entity-wise performance of the six major entities in the dataset. The precision value for ORGANIZATION is very less in Tamil-English text. The recall value of PERSON is less in Hindi-English text. It was also observed that most of the entities were wrongly tagged as O tag.

8 Conclusions

The users in social media communicates to one another by mixing multiple languages. People tend to deliver their views in social media using such code-mixed

language. At business point of view, extracting opinion and discussion about a product from social media text has a greater importance. Extracting entities like Person, Location or Organization from such code-mixed dataset is a tedious task. The dataset provided by the CMEE-IL organizers is from Twitter and other few microblogs for the code-mixed languages like Hindi-English and Tamil-English. Our submission for the task included three different approaches. To perform the entity extraction task, first two approaches used the word embedding features of wang2vec and word2vec. Training of the word embedding models utilized the training data along with some additionally collected dataset. The third system uses only traditional stylometric features. Training and testing of the three approaches were carried out using SVM. The result for the system using stylometric feature was better than the word-embedding based systems. From the results of the task organizers, we can observe that the accuracy of Tamil-English is less and recall is less for all the systems. An entity-wise evaluation for the major entities is also illustrated in this paper. An increase in the size of additional unlabeled data will increase the capability of the word embedding models to capture the syntactic similarity of the sentences, more accurately. Hence, as future work, we are planning to collect more unlabeled data for approaches using word embedding models. We will also be focusing on character based embedding models with deep learning techniques and investigate whether including POS tag information with stylometric features will improve the performance of the system.

References

1. Remmiya Devi, G., Veena, P.V., Anand Kumar, M., Soman, K.P.: AMRITA_CEN@ FIRE 2016: code-mix entity extraction for Hindi-English and Tamil-English tweets. In: CEUR Workshop Proceedings, vol. 1737, pp. 304–308 (2016)
2. Wang, L., Chris, D., Alan, B., Isabel, T.: Two/too simple adaptations of word2vec for syntax problems. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1299–1304 (2015)
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013)
4. Joachims, T.: SVMlight: support vector machine. Cornell University (2008). http://svmlight.joachims.org/
5. Vyas, Y., Gella, S., Sharma, J., Bali, K., Choudhury, M.: POS tagging of English-Hindi code-mixed social media content. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 974–979. Association for Computational Linguistics (2014)
6. Barman, U., Das, A., Wagner, J., Foster, J.: Code mixing: a challenge for language identification in the language of social media. In: Conference on Empirical Methods in Natural Language Processing (EMNLP) 2014, p. 13 (2014)
7. Asnani, K., Pawar, J.D.: Discovering thematic knowledge from code-mixed chat messages using topic model. In: Proceedings of the 3rd Workshop on Indian Language Data: Resources and Evaluation (WILDRE3), pp. 104–109 (2016)

8. Anand Kumar, M., Soman, K.P.: Amrita_CEN@MSIR-FIRE2016: code-mixed question classification using BoWs and RNN embeddings. In: CEUR Workshop Proceedings, vol. 1737, pp. 122–125 (2016)
9. Gupta, D., Shweta, Tripathi, S., Ekbal, A., Bhattacharyya, P.: A hybrid approach for entity extraction in code-mixed social media data. In: CEUR Workshop Proceedings, vol. 1737, pp. 298–303 (2016)
10. Abinaya, N., John, N., Barathi Ganesh, H., Anand Kumar, M., Soman, K.P.: AMRITA-CEN@FIRE-2014: named entity recognition for Indian languages using rich features. In: ACM International Conference Proceeding Series, pp. 103–111 (2014)
11. Rao, P., Devi, S.: CMEE-IL: code mix entity extraction in Indian languages from social media text @ FIRE 2016 an overview. In: CEUR Workshop Proceedings, vol. 1737, pp. 289–295 (2016)
12. Anand Kumar, M., Shriya, S., Soman, K.P.: AMRITA-CEN@FIRE 2015: extracting entities for social media texts in Indian languages. In: CEUR Workshop Proceedings, vol. 1587, pp. 85–88 (2015)
13. Sanjay, S., Anand Kumar, M., Soman, K.P.: AMRITA_CEN-NLP@FIRE 2015:CRF based named entity extraction for Twitter microposts. CEUR Workshop Proceedings, vol. 1587, pp. 96–99 (2015)
14. Pallavi, K., Srividhya, K., Victor, R., Ramya, M.: HITS@FIRE task 2015: Twitter based named entity recognizer for Indian languages. In: CEUR Workshop Proceedings, vol. 1587, pp. 81–84 (2015)
15. Rao, P., Malarkodi, C., Vijay Sundar Ram, R., Devi, S.: ESM-IL: entity extraction from social media text for Indian languages @ FIRE 2015 an overview. In: CEUR Workshop Proceedings, vol. 1587, pp. 74–80 (2015)
16. Bhat, I., Shrivastava, M., Bhat, R.: Code mixed entity extraction in Indian languages using neural networks. In: CEUR Workshop Proceedings, vol. 1737, pp. 296–297 (2016)
17. Skanda, S., Singh, S., Remmiya Devi, G., Veena, P.V., Anand Kumar, M., Soman, K.P.: CEN@Amrita FIRE 2016: context based character embeddings for entity extraction in code-mixed text. In: CEUR Workshop Proceedings, vol. 1737, pp. 321–324 (2016)
18. Basile, P., Caputo, A., Gentile, A.L., Rizzo, G.: Overview of the EVALITA: named entity rEcognition and linking in Italian tweets (NEEL-IT) task. In: CEUR Workshop Proceedings, vol. 1749 (2016)
19. Strauss, B., Toma, B.E., Ritter, A., de Marneffe, M.-C., Xu, W.: Results of the WNUT16 named entity recognition shared task. In: Proceedings of the 2nd Workshop on Noisy User-generated Text, pp. 138–144 (2016)
20. Banerjee, S., Chakma, K., Naskar, S., Das, A., Rosso, P., Bandyopadhyay, S., Choudhury, M.: Overview of the mixed script information retrieval (MSIR) at FIRE-2016. In: CEUR Workshop Proceedings, vol. 1737, pp. 94–99 (2016)
21. Patra, B.G., Das, D., Das, A., Prasath, R.: Shared task on sentiment analysis in indian languages (SAIL) tweets - an overview. In: Prasath, R., Vuppala, A.K., Kathirvalavakumar, T. (eds.) MIKE 2015. LNCS (LNAI), vol. 9468, pp. 650–655. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26832-3_61
22. Shriya, S., Vinayakumar, R., Anand Kumar, M., Soman, K.P.: AMRITA-CEN@SAIL2015: sentiment analysis in Indian languages. In: Prasath, R., Vuppala, A.K., Kathirvalavakumar, T. (eds.) MIKE 2015. LNCS (LNAI), vol. 9468, pp. 703–710. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26832-3_67