CEN@Amrita FIRE 2016: Context based Character Embeddings for Entity Extraction in Code-Mixed Text

Srinidhi Skanda V
Center for Computational Engineering and Networking(CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, Amrita University,India
skanda9051@gmail.com

Shivkaran Singh
Center for Computational Engineering and Networking(CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham
Amrita University,India
shivkaran.ssokhey@gmail.com

Remmiya Devi G
Center for Computational Engineering and Networking(CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham
Amrita University,India

Veena P V
Center for Computational Engineering and Networking(CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, Amrita University,India

Anand Kumar M
Center for Computational Engineering and Networking(CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, Amrita University,India
m_anandkumar@cb.amrita.edu

Soman K P
Centre for Computational Engineering and Networking(CEN)
Amrita School of Engineering Coimbatore.
Amrita Vishwa Vidyapeetham, Amrita University,India

ABSTRACT
This paper presents the working methodology and results on Code Mix Entity Extraction in Indian Languages (CMEE-IL) shared the task of FIRE-2016. The aim of the task is to identify various entities such as a person, organization, movie and location names in a given code-mixed tweets. The tweets in code mix are written in English mixed with Hindi or Tamil. In this work, Entity Extraction system is implemented for both Hindi-English and Tamil-English code-mix tweets. The system employs context based character embedding features to train Support Vector Machine (SVM) classifier. The training data was tokenized such that each line containing a single word. These words were further split into characters. Embedding vectors of these characters are appended with the I-O-B tags and used for training the system. During the testing phase, we use context embedding features to predict the entity tags for characters in test data. We observed that the cross-validation accuracy using character embedding gave better results for Hindi-English twitter dataset compare to Tamil-English twitter dataset.

CCS Concepts
• Information Retrieval → Retrieval tasks and goals → Information Extraction • Machine Learning → Machine Learning approaches → Kernel Methods → Support Vector Machines

Keywords
Named-entity recognition, Information extraction, Code-Mixed Support vector machine (SVM), Word embedding, Context-based character embedding.

1. INTRODUCTION
The explosion of social networking site like Facebook, Twitter and Instagram in a linguistic diverse country like India impacted many users using multiple languages in tweets, personal blogs etc. The scarcity of proper input tools for Indian Languages forced the user to use the roman script mixed with the native script that led to the code mixed language. This code-mixing further complicates the application of Natural Language Processing (NLP) methods. One such application is Named Entity Recognition (NER), which identifies and classifies entities into categories such as a person, organization, date-time etc. The NER is used in text mining application, information retrieval, and machine translation etc. NER task for Indian languages is a complex task. The use of code-mixing further complicates this task.

The shared task on Code Mix Entity Extraction in Indian Languages (CMEE-IL)[1] held at Forum for Information Retrieval (2016) focuses on extracting entities from the code-mix twitter data.

Many works have done on Entity Recognition using word embedding as a feature. Few research works are done on Entity extraction using Character Embedding technique such as Entity recognition using character embedding by Cicero dos Santos et al. [1], Text segmentation using character-level text embedding is done by Grzegorz [2]. Some other works based on code -mix data are SVM-based classification for entity extraction for Indian languages [3]. Entity extraction was done based on Conditional Random Field [4] and Identification and linking of tweets was performed earlier [10].

We have submitted a system based on the character embedding a feature representation. To retrieve character embedding features, the word2vec model is used [5]. Word2vec converts input character to a vector of n-dimension. Feature representation obtained from word2vec is used for training the system. Classifier used for training - Support Vector Machine (SVM) which is based on machine learning [6].

An outline of the task is given in Section 2. Implemented system is described in Section 3. Section 4 describes Experiment and result. The conclusion is discussed in Section 5.

[1]http://www.au-kbc.org/nlp/CMEE-FIRE2016/

2. CODE MIX ENTITY EXTRACTION (CMEE-IL) TASK

Hindi-English and Tamil-English code-mixed language is given as training data, to implement the shared task on CMEE-IL. The task is to extract the named entities along with their named entity tags. The training and testing dataset had raw tweets and annotations in different files for code-mixed Hindi-English and Tamil-English language. There were 18 types of entities present in the training dataset in which entities like a person, location, and entertainment were more in number compare to other entity types. The number of tweets and number of tokens in training, as well as testing corpus for both code-mixed languages, are shown in Table 1.

Table 1. Dataset Count

Tweet Type	# of Tweet		# of Tokens	
	Train	Test	Train	Test
Hindi-English	2700	7429	130866	130868
Tamil-English	3200	1376	41466	20190

The sample tweets from training dataset containing code-mix Hindi-English and Tamil-English can be seen in Table 2. We noticed there were some tweets written completely or partially in the Tamil language in Tamil-English code-mixed corpus. No such findings were noticed in the Hindi-English code-mixed corpus.

Table 2. Examples of code-mixed Tweets

English -Tamil	@_madhumitha_ next time im in chennaidef going over like \"hello ennakuongapethitheriyum\"
	@thoatta நான் சொல்ல வரது டீஸர் டிரைலர் நல்லா இருக்குனு சொன்ன படங்கள எல்லாம் வெற்றி அடையல.over expectations nala flop airumnu STD சொல்லுது
English -Hindi	@aixo_anjum sister me aesy bkwas nhi krta na muje shoq he I liked your pinned tweet at that time too just wait I shall give you proof

Further, we analyzed the corpus for the counts of major entities. The count of major entities with the tags in the training corpus can be seen in Table 3. It can be observed from Table 3 that entities related to Entertainment tags were the most occurring entities.

Table 3. Most frequent entity tags

Code Mix Lang.	Entity Tags			
	Person	Location	Organization	Entertainment
Hindi-English	712	194	109	810
Tamil-English	661	188	68	260

The given raw text file contains Tweet-ID, User-ID and corresponding tweet. The annotation file was arranged in columns, with each column representing Tweet-ID, User-ID, Named Entity (NE) type, Named Entity String, Start Index and Length of the string. The participants were asked to submit a similar annotation file after extraction all the entity related information from test data file.

3. SYSTEM DESCRIPTION

In the proposed system, feature extraction plays a vital role as the accuracy of the system relies majorly on the extracted features. Before extracting any features, the dataset should be in an appropriate format for the learning algorithm. The I-O-B tagging format was used for this format conversion. The first step in format conversion was to tokenize the given data and arrange each word (token) per line. To identify start and end of a tweet we added a tag <S> to make it easy for further processing. The proposed system is based on character based context embedding. The basic intuition behind the character based context embedding is shown in Figure 1. In the figure, the Word(W_0) (LIKE[2]) is split into corresponding characters. Each character is represented by a $50-$ dimensional vector (vector length is user defined). These character embeddings are further concatenated to form a word vector.

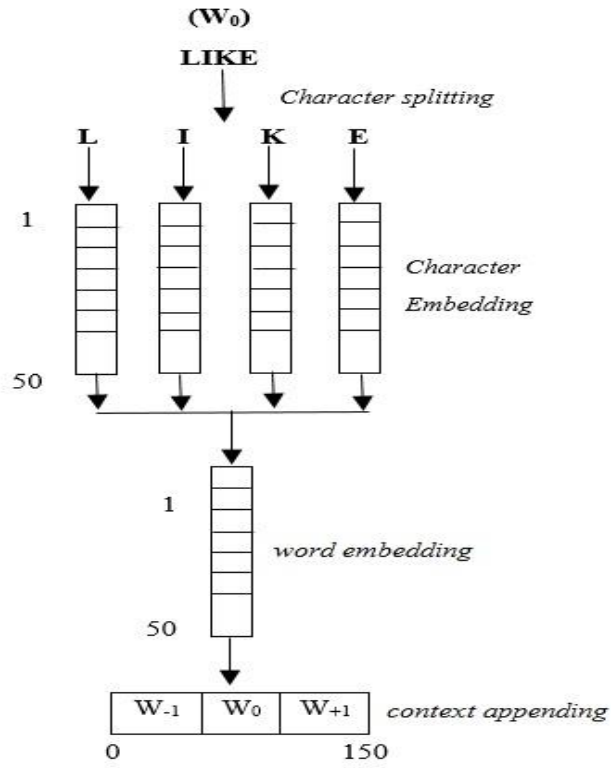

Figure 1. Character Based Context Embedding

This unique vector is appended with the corresponding word. This 50-dimensional representation is converted to 150-dimenstion by appending the neighboring context words. W_{-1} represents previous context word of the word W_0 and W_{+1} represent the next context word of the word W_0. Appending the vectors of W_{-1}, W_0 and W_{+1} are called context appending.

The overall methodology is split into two modules; each module is described in separate figures. Figure 2 shows the method

[2] The capitalization of character is just for representation purpose.

followed for feature representation. Figure 3 shows steps involved in the classification task.

The first module depicted in Figure 2 explains step involved in feature representation. We collected additional datasets from the different data source to improve the accuracy of the system. The additional dataset for Tamil-English was collected from Sentiment Analysis in Indian Languages (SAIL-2015) [9]. Dataset from International Conference on Natural Language Processing (ICON) 2015 POS Tagging task [8], Mixed Script Information Retrieval 2016 (MSIR) [7] and some twitter dataset using web scraping were used for Hindi-English.

This huge file comprising of training as well as additional dataset collected was then subjected for tokenization. These tokenized words are further split into corresponding characters. These characters are fed to vector embedding module. The output of the module will have a matrix of length 1×50 for every character in our corpus. These vectors are concatenated back to words using an appending module. The embedding features for each word are further subjected to context appending, resulting in a vector of length 1×150. These vectors contain the information about the word as well as neighboring word contexts.

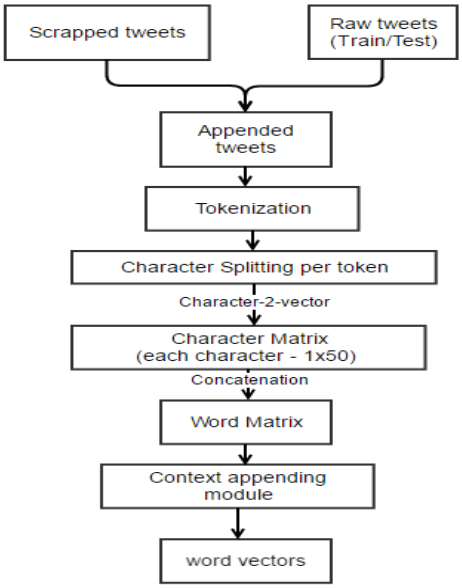

Figure 2. Feature extraction

Format conversion involves raw tweets from the given training data set and annotated file is given to the I-O-B module. I-O-B converts given training data into the I-O-B format. Here I-O-B formatted training set consists of training data along with corresponding I-O-B tags. This formatted training set is appended with the feature vector to form training data that has to be fed to the classifier.

Figure 3 describes procedure involved in classification. During the training phase, the classification model is built based on the feature and label pair. During the testing phase, test data along with its embedding features are fed to predict entity tags for testing data. In Figure 3, Context vectors represent features and label pairs that are fed to SVM module. Here we used SVM Light tool to build anSVM-based classifier model. The classifier takes features and label as an input and trains itself finally builds a classification model. From the learned classification model, the system predicts the entity tags for testing data. Predicted tags extracted from the output file are converted to required annotated format.

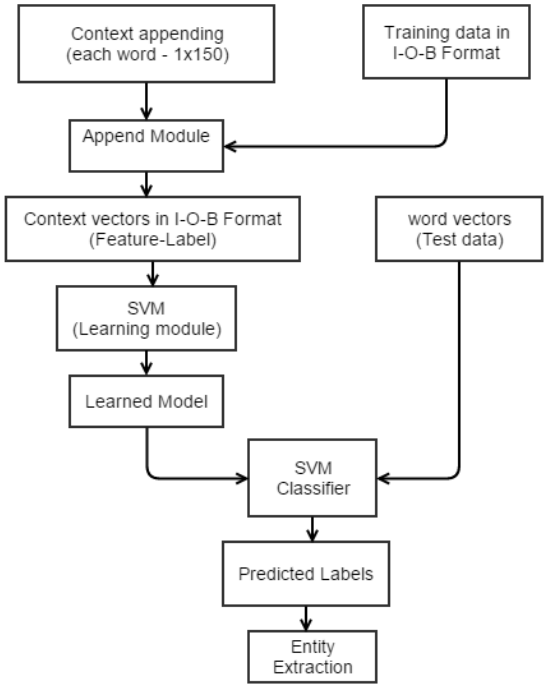

Figure 3. Architecture of the implemented system

4. RESULT AND ANALYSIS

This paper describes work of entity extraction system for code-mix twitter data using character embedding technique. In character embedding approach words in raw code-mix training data is tokenized into one word per line fashion and further split into characters. This character is embedded with the n-dimension vectors to produce feature vectors. Feature vectors appended with the I-O-B tags to represent feature - label sets. This set is fed to Support Vector Machine (SVM) classifier to train a classification model. During testing phase test data undergoes character embedding to represent the test data as a feature vector. These feature vectors are fed to classification model that we already trained. Classification model predicts the entity tags for the test data set. The output of the test data set is converted to a suitable form to represent the Annotated format. Annotated format contains Tweet ID, User- ID; word its corresponding predicted tags, Index and Length. Table 3 shows Cross Validation result for both Hindi-English and Tamil-English datasets. The overall accuracy of character embedding for the Hindi-English system is 95.996%. For Tamil-English datasets overall accuracy of character embedding is 94.3451%.

Table 3. Cross-Validation results for character embedding

Tweet Type	Result			
	Overall accuracy	Known	Ambiguous Known	Unknown
Hindi-English	95.9962	97.6447	83.8945	92.1155
Tamil-English	94.3451	95.8483	91.7017	89.6086

Table 5 and 6 displays result participated in a shared task. Implemented system ranked eighth in Hindi-English twitter dataset with Precision 48.17 and F-measure 32.83. For Tamil-English our system was ranked fifth with Precision 47.62 and F-measure 20.94.

Table 5. Result of Hindi-English twitter data

Rank	Team	Best run		
		Precision	Recall	F-Measure
1	Irshad-IIIT-Hyd	80.92	59.00	68.24
2	Deepak-IIT-Patna	81.15	50.39	62.17
3	Veena-Amrita-T1	79.88	41.37	54.51
	CEN@Amrita	**48.17**	**24.90**	**32.83**

Table 6. Result of Tamil-English twitter data

Rank	Team	Best run		
		Precision	Recall	F-Measure
1	Deepak-IITPatna	79.92	30.47	44.12
2	VeenaAmrita-T1	79.51	21.88	34.32
3	BharathiAmritha-T2	79.56	19.59	31.44
	CEN@Amrita	**47.62**	**13.42**	**20.94**

5. CONCLUSION AND FUTURE SCOPE

The proposed system for code-mix entity extraction is submitted as a part shared task on code-mix entity extraction system in Indian languages (CMEE-IL) conducted by FIRE 2016. Task involves extracting entity from the code-mix tweets. Given dataset consist of Hindi-English and Tamil-English code-mix tweets. In our work, we implemented character embedding system. We conclude that overall accuracy of Character embedding for Hindi-English is better compared to the Tamil-English.

Few errors that resulted in decrease performance of the system are

1. While splitting words to characters due to encoding problem and noise present in the dataset some characters are not properly represented.
2. The performance of the implemented system could be improved by using RNN or CNN based models.

6. ACKNOWLEDGMENT

We would like to give thanks to the task organizer - Forum for Information Retrieval Evaluation. We also thank organizers of CMEE-IL task.

7. REFERENCES

[1] Dos Santos, Cıcero, Victor Guimaraes, R. J. Niterói and Rio de Janeiro. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, p. 25.

[2] Chrupała, Grzegorz. 2013. Text segmentation with character-level text embeddings. *arXiv preprint arXiv:1309.4628*

[3] Anand Kumar, M., Shriya, S., and Soman, K.P. 2015. AMRITA-CEN@FIRE 2015: Extracting entities for social media texts in Indian languages. *CEUR Workshop Proceedings*, 1587:85–88.

[4] Sanjay, S., Anand Kumar, M., and Soman, K.P. 2015. AMRITA-CEN-NLP@FIRE 2015:CRF based named entity extraction for Twitter microposts. *CEUR Workshop Proceedings*, 1587:96–99.

[5] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

[6] Joachims and Thorsten. 1999. SVM-light: Support vector machine. *SVM-Light Support Vector Machine http://svmlight. Joachim. org/, University of Dortmund*, 19(4).

[7] Shared task on mixed script information retrieval, https://msir2016.github.io, 2016.

[8] Vyas, Y., Gella, S., Sharma, J., Bali, K., and Choudhury. October 2014. POS tagging of English-Hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979.

[9] Patra, B.G., Das, D., Das, A., and Prasath, R. 2015. Shared task on sentiment analysis in Indian languages (sail) tweets-an overview. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 650–655.

[10] Barathi Ganesh, H. B., Abinaya, N., Anand Kumar, M., Vinayakumar, R., and Soman, K.P. 2015. AMRITA-CEN@ NEEL: Identification and Linking of Twitter Entities. *Making Sense of Microposts (# Microposts2015)*

[11] Saha, Sujan Kumar, Sanjay Chatterji, Sandipan Dandapat, Sudeshna Sarkar, and Pabitra Mitra. 2008. A hybrid approach for named entity recognition in indian languages. In *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages*, pp. 17-24.

[12] Abinaya, N., Neethu John., Barathi HB Ganesh., Anand M. Kumar., and Soman, K.P. 2014. AMRITA_CEN@ FIRE-2014: Named Entity Recognition for Indian Languages using Rich Features. In *Proceedings of the Forum for Information Retrieval Evaluation*, pp. 103-111.

[13] Xue, Bai, Chen Fu, and Zhan Shaobin. 2014. A study on sentiment computing and classification of sinaweibo with word2vec. In 2014 *IEEE International congress on Big Data*, pp. 358-363.

[14] Le, Quoc, V., and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*, vol. 14, pp. 1188-1196.