Named Entity Recognition for Code Mixing in Indian Languages using Hybrid Approach

Rupal Bhargava[1] Bapiraju Vamsi Tadikonda [2] Yashvardhan Sharma[3]

WiSoc Lab, Department of Computer Science
Birla Institute of Technology and Science, Pilani Campus
Pilani-333031

{rupal.bhargava[1], f2013039[2], yash[3]} @pilani.bits-pilani.ac.in

ABSTRACT

Automating the process of Named Entity Recognition has received a lot of attention over past few years in Social Media Text. Named Entities are real world objects such as Person, Organization, Product, Location. Identifying these entities in social media text is an important challenging task due the informal nature of text present on social media. One such challenge that is faced in recognizing named entities in Indian Social Media Text is Code Mixing. Code Mixing is usage of more than one language in a sentence. Being a multilingual country, people of India tend to know more than one language, which in turn results in the code mixing of text while expressing their opinions. This paper describes the proposed approach for shared task CMEE-IL (Code Mix Entity Extraction in Indian Language), FIRE 2016. Proposed algorithm uses a hybrid approach of a dictionary cum supervised classification approach for identifying entities in Code Mix Text of Indian Languages such as Hindi- English and Tamil-English.

CCS Concepts

•**Computing methodologies → Natural language processing;** *Information extraction; Language resources; Machine learning;*

Keywords

Code Mixing, Indian Languages, Named Entity Recognition, Natural Language Processing, Information Retrieval

1. INTRODUCTION

India is a multilingual country with more than 1600 languages being spoken such as Hindi, Punjabi,Bengali, Telugu, Marathi, Tamil, Gujarati and many more. With the introduction of Indic keyboards and articles that use Indic languages, people started using Indic languages in their normal conversation and this has made people to converse easily on the internet. With such a scenario, it is common that people know at least one more language apart from their native language due to which there is a high possibility that people mix words from two or more different languages while writing or speaking something. This mixing of words in sentences is referred as code mixing. Code mixing is present where people speak in an informal way, like in social media.

Growing usage of social media platforms like Facebook, Twitter and WhatsApp has led to an increase of code mix data present because of interplay of Indian languages. Hence for making best use of this data it needs to be analyzed. Indic languages are used very much nowadays in general conversations but there are a few Natural Language Processing (NLP) resources that are available.When code mixed text is combined as well very less resources are present. Hence there is a greater need for developing NLP tool that can handle code mixed texts with Indic languages.

Entity recognition is a very important subtask of Information extraction and find its applications in information retrieval, machine translation and other higher NLP applications such as coreference resolution. Named Entities are names of famous persons, organizations, locations, animals etc.Named entities have a many uses like in sentiment analysis,where recognizing named entities is important as they don't add much value to the statement. Similarly while tagging articles named entities are required for better search results. There are many such uses where named entities are used so named entity recognition is very important.Towards this, FIRE 2016 has organized a task for entity recognition in code mix text for Indian languages which identifies named entity in code mix text of English-Hindi and Tamil- English code mixed tweets. The Task was to identify the various entities such as person names, organization names, movie names, location names in a given tweet.

Rest of the paper is organized as follows. Section 2 explains the related work that has been done in the past few years. Section 3 presents the analysis of data set provided by CMEE-IL 2016 Task Organizers. Section 4 explains the Proposed Technique that have been performed for the task with block diagrams. Section 5 discusses algorithm to explain the procedure. Section 6 elaborates the evaluation and experimental results and error analysis. Section 7 concludes the paper and presents future work.

2. RELATED WORK

In Named Entity recognition there has been significant research done so far in English but same cannot be said for Indian Languages due to rich morphology of indian languages. Sujan et al [10] proposed a Named Entity Recognition (NER) system which is a hybrid of maximum entropy model, language specific rules and gazetteer lists.This system performs well for hindi and bengali languages. Malarkodi et al [5] have developed a system specific for tamil language using Conditional random fields(CRF). Entity Extraction from Social Media Text - Indian Languages (ESM-IL) task of FIRE 2015 [8] had proposed a task of identifying Entities in Indian Languages for social media text. As a baseline system, task organizers build a system which just used raw data and was trained on Conditional Random Field. It was observed by them that most of the participants obtained similar precision as that of baseline. However, there was a significant improvement of recall over the baseline. As a submission to the task Pallavi et al [6] proposed a system using Conditional Random Fields(CRF)whereas Anand et al [1] proposed a system using a Support Vector Machine (SVM). Kamal et al. [11] used POS tag as a state and developed a system using Hidden Markov Model (HMM) Classifier for the task.

In current scenario there has been a lot of work going on recently found out trend of code-mixed texts in Indian Languages. Parth et al. [4] in 2014 formally introduced the concept of Mixed Script Information Retrieval (MSIR) and challenges associated with it. Code Mixing recieved attained some attention in 2015. In shared task MSIR, FIRE 2015 [9] problem was proposed, for identifying mix scripts in text along with 9 different Indian languages, Named Entities and punctuations for which significant results were obtained. It was found that most confusing language pair was that of Hindi and Gujarati. Further, it was concluded that performance of the system for each of the category is dependent on the tokens used for that category. Not only this code mix has found its application in different areas such as Query Labeling [3], Sentiment Analysis [2], Question Classification etc.

3. DATA ANALYSIS

Data Set provided by task organizers contained two code mix data set, Tamil-English and Hindi- English. In Each dataset , the training data consisted of two files, A text file containing raw tweets along with their tweetID and UserID and another text file containing Annotations to the tweets present in the raw tweets file. The raw tweet files consists of 2700 tweets in the Hindi-English corpus and 3200 tweets in the Tamil-English corpus. All the tweets in the Hindi-English corpus were already romanized whereas the tamil-english corpus had a mixture of both tamil script and romanised script. There were 22 tags present in the corpus as mentioned in Table 1.

Named Entity (NE) Tag Person, Entertainment and Location occupies majority of the instances in Tamil -English corpus. Person tag comprises of Names of Famous Actors, Actresses, Politicians, New Reporters and Social Media Celebrities. Entertainment Comprises of Names of Famous TV shows and movies while Location consists of Names of Famous Cities, Indian Towns and Names of Countries.

Apart from these, there are some Numerical and Time based Tags that are present as well which comprise of the remaining part of Tamil-English data set. These tags include Count, Distance, date, money, month, time and year. Money represents numbers along with a monetary tag like '15 dollars'. Organization is another tag present which consists of names of organizations. The rest of the tags have very less annotations present in the annotated file.

Table 1: Frequency of NE in both Data Sets

Type of NE	Tamil-English	Hindi-English
Artifact	18	25
Count	94	132
Date	14	33
Disease	5	7
Distance	4	0
Entertainment	260	810
Facilities	23	10
Livthings	16	7
Location	188	194
Locomotive	5	13
Materials	28	24
Money	66	25
Month	25	10
Organization	68	109
Period	53	44
Person	661	712
Plants	3	1
Quantity	0	2
Sday	6	23
Time	18	22
Year	54	143
	1609	2346

Comparing Hindi-English with Tamil-English, The percentage of tags for the minority tags remains almost the same. But in the Hindi- English corpus the Tag Entertainment has the highest number of annotations present.It is followed by Person which is close to Entertainment.The rest of order remains the same but with varying percentages.

4. PROPOSED TECHNIQUE

A word level NE-recognition system is designed to recognise Named Entities in a tweet. The proposed methodology involves a pipelined approach for detecting each NE tag and has been divided into following four phases:

1. Pre-processing
2. Number Based Named Entity Recognition
3. Gazetteer List Based Named Entity Recognition
4. Tree Based Named Entity Identifier

4.1 Pre-Processing

The Data is pre-processed before detecting named entities. This is done to ensure that the data is uniform and the system can benefit from that. The preprocessing consists of creating a copy of the string in lowercase for uniformity. It also removes all links present in the tweet. This preprocessed tweet along with the original tweet is then passed to the next phase.

4.2 Number Based Entity Recognition

This phase of proposed algorithm identifies number based entity such as date, time, month, day, year, money, period,

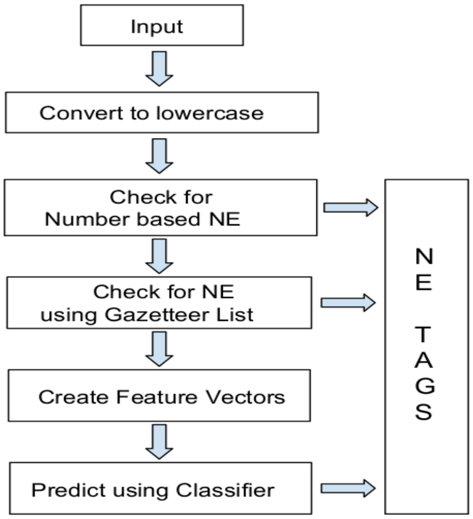

Figure 1: Block Diagram for Proposed Algorithm

quantity, distance and count using a set of Regular Expressions Regular expressions are designed based on the common patterns observed in the annotations for these tags. Regular expressions work best in detection for these tags because there are limited variations possible for each of these Tags. For example the tag Day can be only one of the 7 possible days of a week in a language. So detecting them using regular expressions will be efficient. While checking for NE tags there is a possibility of having multiple tags attached to the same token. To remove ambiguity proposed technique checks for tags in a particular pre defined order.

4.3 Gazetteer Based Entity Recognition

As shown in Table 1, apart from Entertainment, Location, Person and Organization, Rest of the Tags contain very less data that cannot be used to train a classifier. So Gazetteer Lists are used for identifying Named Entities with insufficient training data. Gazetteer lists are created from the annotations given to the training data. While checking in gazetteer list # and @ symbols were ignored.

4.4 Classification

The rest of the NE Tags are identified by creating feature vector for each token of a tweet. These feature vectors are then trained using a decision tree and extremely randomized tree classifier . The features considered for building feature vector are mentioned in Table 2.

Table 2: Features used for creating feature vector.

Sno	Features
1	Presence of token in English dictionary
2	Prefixes of length 1 to 3
3	Suffixes of length 1 to 3
4	Capitalization related features like starting letter capital, all letters capital, other letters capital.
5	Features based on presence or absence of special characters like #, @, numbers, other symbols.
6	Presence of emoticons
7	Token present in gazetteer list.
8	Is previous token a NE Tag.

English dictionary feature is used to identify the presence and absence of an english word i.e if it is a english word it is 1 and if it is a non english word then its 0. Python dictionary called pyenchant [1] was used for this identifying this feature. Also for prefix suffix feature a dictionary was built using most common prefixes and suffixes and presence of these prefixes and suffixes (length = 1 to 3) in tokens were identified using the same. Gazetteer list feature checks for the presence of the token in gazetteers list of the remaining tags and uses this as a feature. Previous token tag was also taken into account to check structure of the tweet. Using all features mentioned in Table 2, Decision tree and Extremely randomized trees are trained for classification [7].

5. ALGORITHM

Algorithm 1 explains the proposed technique for Named Entity Recognition of code mixed text. The System first pre-processes the input by removing the website and twitter links (implemented by callable: Link_remover) and then converts the tweet into lowercase (implemented by callable: Case_conversion). We check for all numerical features like date, time, money, quantity, period, distance, day and count (implemented by callable: check_Numerical).Before adding it to the final predictions we check for overlapping Tags and remove them (using add_without_repetition). This is the second phase of the system.

In the third phase, we tokenize the tweets (using the function Tokenize) and check if any of the token is present in any of the gazetteer list (using check_gazetteer_List) and add them to the final list of tags of that tweet.This ends the third phase of the system. In the final phase we create feature vectors for each tweet and then predict using the classifier (clf) already trained using a training data. The classifier(clf) used are decision trees and extremely randomized trees.

Algorithm 1 Algorithm for Identifying Named Entity Recognition for Code Mixed Text in Indian Language

```
Input: Code-Mixed tweets list , S
Output: Predicted Named Entity Labels, P
Initialization: P=[], toks=[], NE_ Data=[], NE_ Tags=[]
for i=0 to S.length do
    Link_remover(S[i])
    Case_conversion(S[i])
end for
for i=0 to S.length do
    d=check_Numerical(S[i])
    add_without_repetition(d)
    tok =Tokenize(S[i])
    for j=0 to tok.length do
        g = check_gazetteer_List(tok[ j ] )
        add_without_repetition(P , g )
        f = Create_Feature_vector(tok[ j ])
        c=clf.predict(f)
        add_without_repetition(P,c)
    end for
end for
```

[1] http://packages.python.org/pyenchant/

Table 3: Different Versions of Proposed System

Version	Tags trained on Classifier	Classifier Used
1	Person, Entertainment, Location, Organization	Decision Tree
2	Person, Entertainment, Location, Organization	Extremely Randomized Tree
3	Person, Entertainment, Location, Organization, Artifact, Facilities	Decision Tree
4	Person, Entertainment, Location, Organization, Artifact, Facilities	Extremely Randomized Tree

Table 4: Results for Hindi English Proposed System

Runs	Precision	Recall	F-Measure
Run 1	58.66	32.93	42.18
Run 2	58.84	35.32	44.14
Run 3	59.15	34.62	43.68

6. EXPERIMENTS & RESULTS

CMEE-IL, FIRE 2016, had proposed the task of Named Entity Recognition in Hindi-English and Tamil-English Code mixed text. Three runs were submitted for the task evaluation for each language pair. Four versions were created for different runs submitted. In all the versions, numerical feature were detected using numerical function as explained in section 4.2. Rest of the tags were classified using different versions created as specified in Table 3.

The rest of the tags were classified using gazetteer list phase as mentioned in section 4.3.This was done due to low amount of training data available for few tags such as plants, disease, locomotive etc as mentioned in Table 1. All the Variations for the proposed algorithm were evaluated using F-Score for each language pair and finally Run1, 2, 3 for Hindi-English used versions 1, 2 and 4 respectively whereas Run 1, 2, 3 for Tamil-English used versions 1, 3 and 4 respectively.

6.1 Evaluation & Discussion

As shown in Table 4, run 2 performed the best for Hindi-English Proposed System. Similarly, run 2 performed best for Tamil-English as well as indicated in Table 5. Based on the F-Score, it can be concluded that algorithm with more gazetteer List and Extremely Randomized forest (Version 2) performed well in case of Hindi- English. But in case of Tamil-English, algorithm with less Gazetteer List and Decision Tree (Version 3) proved to be effective.

Precision value could be less because of string matching with elements of the gazetteer lists. Also, it can observed that recall value is low for all the runs, this could be due to less number of Named Entities for any sentence which in turn reduces the average recall value.The recall might have increased if partial identification of NE were considered. The proposed system stood fifth among the Hindi-English Systems and was ranked fourth in the case of Tamil-English as shown in Figure 2 & 3 respectively.

Table 5: Results for Tamil English Proposed System

Runs	Precision	Recall	F-Measure
Run 1	55.86	10.87	18.20
Run 2	58.71	12.21	20.22
Run 3	58.94	11.94	19.86

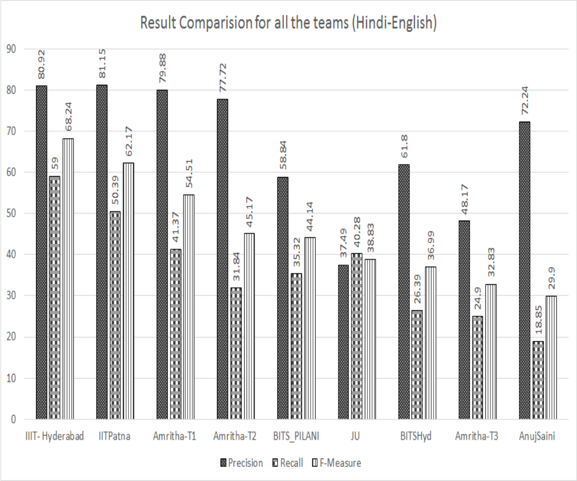

Figure 2: Result Comparision for all teams(Hindi-English)

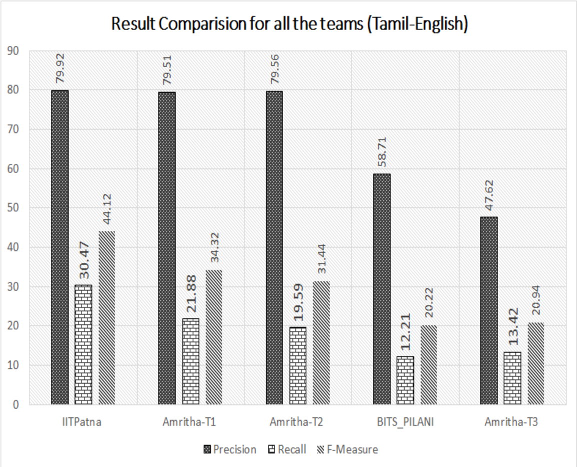

Figure 3: Result Comparision for all teams(Tamil-English)

6.2 Error Analysis

Few phases in proposed approach might have attributed to misclassification for few tags. One such phase can be the gazetteer list phase of the proposed method which is a dictionary based approach and has disadvantages associated with it. When there is less data present in the dictionary the precision will be low for the system. So there is a need for more elements in the list for a better recognition system. Also, if there is an ambiguity in tags then there a chance of misclassification. For example if we say there is a token 'Honey' it can represent a Person like 'Honey Singh' or as a tag material. This ambiguity can only be solved using a classifier.

7. CONCLUSION & FUTURE WORK

In this paper, a hybrid approach of a dictionary cum supervised classification approach for identifying entities in Code Mix Text of Indian Languages such as Hindi- English and Tamil-English is submitted for the task CMEE-IL,FIRE 2016. The proposed system used a pipelined approach to identify the named entities. There are four variants of the system based on the number of tags, the classifier can detect and the classifier used. Further improvisation can be done by incorporating features related to the structure of the sentence. POS Tagging and Chunking of the tweets has not been included, this an be another improvisation in the proposed algorithm. Although we need better POS Taggers for Code mixed Languages which can tag both romanised and non-romanised tweets for the same.

8. REFERENCES

[1] Anand Kumar M, Shriya Se, S. K. Amrita_cen@ fire 2015: Extracting entities for social media texts in indian languages. In *Working notes of FIRE 2015 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December, 2015* (2015), vol. 1587 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 87–90.

[2] Bhargava, R., Sharma, Y., and Sharma, S. Sentiment analysis for mixed script indic sentences. In *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on* (2016), IEEE, pp. 524–529.

[3] Bhargava, R., Sharma, Y., Sharma, S., and Baid, A. Query labelling for indic languages using a hybrid approach. In *Working notes of FIRE 2015 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December, 2015* (2015), vol. 1587 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 40–42.

[4] Gupta, P., Bali, K., Banchs, R. E., Choudhury, M., and Rosso, P. Query expansion for mixed-script information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (2014), ACM, pp. 677–686.

[5] Malarkodi, C., Pattabhi, R., and Sobha, L. D. Tamil ner–coping with real time challenges. In *24th International Conference on Computational Linguistics* (2012), p. 23.

[6] Pallavi, K., Srividhya, K., and Rexiline Ragini John Victor, R. M. Hits@ fire task 2015: Twitter based named entity recognizer for indian languages. In *Working notes of FIRE 2015 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December, 2015* (2015), vol. 1587 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 83–86.

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research 12*, Oct (2011), 2825–2830.

[8] Rao, P. R., Malarkodi, C., and Devi, S. L. Esm-il: Entity extraction from social media text for indian languages@ fire 2015–an overview. In *Working notes of FIRE 2015 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December, 2015* (2015), vol. 1587 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 76–82.

[9] Royal Sequiera, Choudhury, M., Gupta, P., Rosso, P., Kumar, S., Banerjee, S., Naskar, S. K., Bandyopadhyay, S., Chittaranjan, G., Das, A., and Chakma, K. Overview of fire-2015 shared task on mixed script information retrieval. In *Working notes of FIRE 2015 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December, 2015* (2015), vol. 1587 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 19–25.

[10] Saha, S. K., Chatterji, S., Dandapat, S., Sarkar, S., and Mitra, P. 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* (2008), pp. 17–24.

[11] Sarkar, K. A hidden markov model based system for entity extraction from social media english text at fire 2015. In *Working notes of FIRE 2015 - Forum for Information Retrieval Evaluation, Gandhinagar, India, December, 2015* (2015), vol. 1587 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 91–97.