Corpus Creation and Analysis for Named Entity Recognition in Telugu-English Code-Mixed Social Media Data

Vamshi Krishna Srirangam, Appidi Abhinav Reddy, Vinay Singh, Manish Shrivastava
Language Technologies Research Centre (LTRC)
Kohli Centre on Intelligent Systems(KCIS)
International Institute of Information Technology, Hyderabad, India.
{v.srirangam, abhinav.appidi, vinay.singh}@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

Named Entity Recognition(NER) is one of the important tasks in Natural Language Processing(NLP) and also is a sub task of Information Extraction. In this paper we present our work on NER in Telugu-English code-mixed social media data. Code-Mixing, a progeny of multilingualism is a way in which multilingual people express themselves on social media by using linguistics units from different languages within a sentence or speech context. Entity Extraction from social media data such as tweets(twitter)[1] is in general difficult due to its informal nature, code-mixed data further complicates the problem due to its informal, unstructured and incomplete information. We present a Telugu-English code-mixed corpus with the corresponding named entity tags. The named entities used to tag data are Person('Per'), Organization('Org') and Location('Loc'). We experimented with the machine learning models Conditional Random Fields(CRFs), Decision Trees and Bidirectional LSTMs on our corpus which resulted in a F1-score of 0.96, 0.94 and 0.95 respectively.

1 Introduction

People from Multilingual societies often tend to switch between languages while speaking or writing. This phenomenon of interchanging languages is commonly described by two terms "code-mixing" and "code-switching". Code-Mixing refers to the placing or mixing of various linguistic units such as affixes, words, phrases and clauses from two different grammatical systems within the same sentence and speech context. Code-Switching refers to the placing or mixing of units such as words, phrases and sentences from two codes within the same speech context. The structural difference between code-mixing and code-switching can be understood in terms of the position of altered elements. Intersentential modification of codes occurs in code-switching where as the modification of codes is intrasentential in code-mixing. Bokamba (1988). Both code-mixing and code-switching can be observed in social media platforms like Twitter and Facebook, In this paper, we focus on the code-mixing aspect between Telugu and English Languages. Telugu is a Dravidian language spoken majorly in the Indian states of Andhra Pradesh and Telangana. A significant amount of linguistic minorities are present in the neighbouring states. It is one of six languages designated as a classical language of India by the Indian government

The following is an instance taken from Twitter depicting Telugu-English code-mixing, each word in the example is annotated with its respective Named Entity and Language Tags ('Eng' for English and 'Tel' for Telugu).

T1 : *"Sir/other/Eng Rajanna/Person/Tel Siricilla/Location/Tel district/other/Eng loni/other/Tel ee/other/Tel government/other/Eng school/other/Eng ki/other/Tel computers/other/Eng fans/other/Eng vochi/other/Tel samvastharam/other/Tel avthunna/other/Tel Inka/other/Tel permanent/other/Eng electricity/other/Eng raledu/other/Tel Could/other/Eng you/other/Eng please/other/Eng respond/other/Eng @KTRTRS/person/Tel @Collector_RSL/other/Eng"*

Translation: *"Sir it has been a year that this government school in Rajanna Siricilla district has got computers and fans still there is no permanent electricity, Could you please respond @KTRTRS @Collector_RSL "*

[1] https://twitter.com/

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 183–189
Florence, Italy, July 28 - August 2, 2019.

2 Background and Related work

There has been a significant amount of research done in Named Entity Recognition(NER) of resource rich languages Finkel et al. (2005), English Sarkar (2015), German Tjong Kim Sang and De Meulder (2003), French Azpeitia et al. (2014) and Spanish Zea et al. (2016) while the same is not true for code-mixed Indian languages. The FIRE(Forum for Information Retrieval and Extraction)[2] tasks have shed light on NER in Indian languages as well as code-mixed data. The following are some works in code-mixed Indian languages. Bhargava et al. (2016) proposed an algorithm which uses a hybrid approach of a dictionary cum supervised classification approach for identifying entities in Code Mixed Text of Indian Languages such as Hindi- English and Tamil-English.

Nelakuditi et al. (2016) reported work on annotating code mixed English-Telugu data collected from social media site Facebook and creating automatic POS Taggers for this corpus, Singh et al. (2018a) presented an exploration of automatic NER of Hindi-English code-mixed data, Singh et al. (2018b) presented a corpus for NER in Hindi-English Code-Mixed along with experiments on their machine learning models. To the best of our knowledge the corpus we created is the first Telugu-English code-mixed corpus with named entity tags.

3 Corpus and Annotation

The corpus created consists of code-mixed Telugu-English tweets from Twitter. The tweets were scrapped from Twitter using the Twitter Python API[3] which uses the advanced search option of Twitter. The mined tweets are from the past 2 years and belong to topics such as politics, movies, sports, social events etc.. The Hashtags used for tweet mining are shown in the appendicies section. Extensive Pre-processing of tweets is done. The tweets which are noisy and useless i.e contain only URL's and hash-tags are removed. Tokenization of tweets is done using Tweet Tokenizer. Tweets which are written only in English or in Telugu Script are removed too. Finally the tweets which contain linguistic units from both Telugu and English language are considered. This way we made sure that the tweets are Telugu-English code-mixed. We have retrieved a total of 2,16,800 tweets using the python twitter API and after the extensive cleaning we are left with 3968 code-mixed Telugu-English Tweets. The corpus will be made available online soon. The following explains the mapping of tokens with their respective tags.

[2]http://fire.irsi.res.in/fire/2018/home

[3]https://pypi.python.org/pypi/twitterscraper/0.2.7

3.1 Annotation: Named Entity Tagging

We used the following three Named Entities(NE) tags "Person", "Organization" and "Location" to tag the data. The Annotation of the corpus for Named Entity tags was manually done by two persons with linguistic background who are well proficient in both Telugu and English. Each of three tags("Person", "Organization" and "Location") is divided into B-tag (Beginner tag) and I-tag (Intermediate tag) according to the BIO standard. Thus we have now a total of six tags and an 'Other' tag to indicate if it does not belong to any of the six tags. The B-tag is used to tag a word which is the Beginning word of a Named Entity. I-tag is used if a Named Entity is split into multiple continuous and I-tag is assigned to the words which follow the Beginning word. The following explains each of the six tags used for annotation.

The 'Per' tag refers to the 'Person' entity which is the name of the Person, twitter handles and nicknames of people. The 'B-Per' tag is given to the Beginning word of a Person name and 'I-Per' tag is given to the Intermediate word if the Person name is split into multiple continuous.

The 'Org' tag refers to 'Organization' entity which is the name of the social and political organizations like 'Hindus', 'Muslims', 'Bharatiya Janatha Party', 'BJP', 'TRS' and government institutions like 'Reserve Bank of India'. Social media organizations and companies like 'Twitter', 'facebook', 'Google'. The 'B-Org' tag is given to the beginning word of a Organization name and the 'I-Org' tag is given to the Intermediate word of the Organization name, if the Organization name is split into multiple continuous.

The 'Loc' tag refers to 'Location' entity which is the name of the places like 'Hyderabad', 'USA', 'Telangana', 'India'. The 'B-Loc' tag is given to the Beginning word of the Location name and 'I-Loc' tag is given to the Intermediate word of a

	Cohen Kappa
B-Loc	0.97
B-Org	0.95
B-Per	0.94
I-Loc	0.97
I-Org	0.92
I-Per	0.93

Table 1: Inter Annotator Agreement.

Tag	Count of Tokens
B-Loc	5429
B-Org	2257
B-Per	4888
I-Loc	352
I-Org	201
I-Per	782
Total NE tokens	13909

Table 2: Tags and their Count in Corpus

Location name, if the Location name is split into multiple continuous.
The following is an instance of annotation.

T2 : *"repu/other Hyderabad/B-Loc velli/other canara/B-Org bank/I-Org main/other office/other lo/other mahesh/B-Per babu/I-per ni/other meet/other avudham/other "*

Translation: "we will meet mahesh babu tomorrow at the canara bank main office in Hyderabad"

3.2 Inter Annotator Agreement

The Annotation of the corpus for NE tags was done by two persons with linguistic background who are well proficient in both Telugu and English. The quality of the annotation is validated using inter annotator agreement(IAA) between two annotation sets of 3968 tweets and 115772 tokens using Cohen's Kappa coefficient Hallgren (2012). The agreement is significantly high. The agreement between the 'Location' tokens is high while that of 'Organization' and 'Person' tokens is comparatively low due to unclear context and the presence of uncommon or confusing person and organization names. Table 1 shows the Inter annotator agreement.

4 Data statistics

We have retrieved 2,16,800 tweets using the python twitter API. we are left with 3968 code-mixed Telugu-English Tweets after the extensive cleaning. As part of the annotation using six named entity tags and 'other' tag we tagged 115772 tokens. The average length of each tweet is about 29 words. Table 9 shows the distribution of tags.

5 Experiments

In this section we present the experiments using different combinations of features and systems. In order to determine the effect of each feature and parameters of the model we performed several experiments using some set of features at once and all at a time simultaneously changing the parameters of the model, like criterion ('Information gain', 'gini') and maximum depth of the tree for decision tree model, regularization parameters and algorithms of optimization like 'L2 regularization'[4], 'Avg. Perceptron' and 'Passive Aggressive' for CRF. Optimization algorithms and loss functions in LSTM. We used 5 fold cross validation in order to validate our classification models. We used 'scikit-learn' and 'keras' libraries for the implementation of the above algorithms.

Conditional Random Field (CRF) : Conditional Random Fields (CRF's) are a class of statistical modelling methods applied in machine learning and often used for structured prediction tasks. In sequence labelling tasks like POS Tagging, adjective is more likely to be followed by a noun than a verb. In NER using the BIO standard annotation, I-ORG cannot follow I-PER. We wish to look at sentence level rather than just word level as looking at the correlations between the labels in sentence is beneficial, so we chose to work with CRF's in this problem of named entity tagging. We have experimented with regularization parameters and algorithms of optimization like 'L2 regularization', 'Avg. Perceptron' and 'Passive Aggressive' for CRF.

Decision Tree : Decision Trees use tree like structure to solve classification problems where the leaf nodes represent the class labels and the internal nodes of the tree represent attributes. We

[4]https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c

have experimented with parameters like criterion ('Information gain', 'gini') and maximum depth of the tree. Pedregosa et al. (2011)

BiLSTMs : Long short term memory is a Recurrent Neural Network architecture used in the field of deep learning. LSTM networks were first introduced by Hochreiter and Schmidhuber (1997) and then they were popularized by significant amount of work done by many other authors. LSTMs are capable of learning the long term dependencies which help us in getting better results by capturing the previous context. We have BiLSTMs in our experiments, a BiLSTM is a Bi-directional LSTM in which the signal propagates both backward as well as forward in time. We have experimented with Optimization algorithms and loss functions in LSTM.

5.1 Features

The features to our machine learning models consists of character, lexical and word level features such as char N-Grams of size 2 and 3 in order to capture the information from suffixes, emoticons, social special mentions like '#', '@' patterns of punctuation, numbers, numbers in the string and also previous tag information, the same all features from previous and next tokens are used as contextual features.

1. **Character N-Grams:** N-gram is a contiguous sequence of n items from a given sample of text or speech, here the items are characters. N-Grams are simple and scalable and can help capture the contextual information. Character N-Grams are language independent Majumder et al. (2002) and have proven to be efficient in the task of text classification. They are helpful when the text suffers from problems such as misspellings Cavnar et al. (1994); Huffman (1995); Lodhi et al. (2002). Group of chars can help in capturing the semantic information and especially helpful in cases like ours of code-mixed language where there is an informal use of words, which vary significantly from the standard Telugu-English words.

2. **Word N-Grams:** We use word N-Grams, where we used the previous and the next word as a feature vector to train our model which serve as contextual features. Jahangir et al. (2012)

3. **Capitalization:** In social media people tend to use capital letters to refer to the names of the persons, locations and orgs, at times they write the entire name in capitals von Däniken and Cieliebak (2017) to give special importance or to denote aggression. This gives rise to a couple of binary features. One feature is to indicate if the beginning letter of a word is capital and the other to indicate if the entire word is capitalized.

4. **Mentions and Hashtags:** In social media organizations like twitter, people use '@' mentions to refer to persons or organizations, they use '#' hash tags in order to make something notable or to make a topic trending. Thus the presence of these two gives a good probability for the word being a named entity.

5. **Numbers in String:** In social media, we can see people using alphanumeric characters, generally to save the typing effort, shorten the message length or to showcase their style. When observed in our corpus, words containing alphanumeric are generally not named entities. Thus the presence of alphanumeric in words helps us in identifying the negative samples.

6. **Previous Word Tag:** Contextual features play an important role in predicting the tag for the current word. Thus the tag of the previous word is also taken into account while predicting the tag of the current word. All the I-tags come after the B-tags.

7. **Common Symbols:** It is observed that currency symbols, brackets like '(', '[', etc and other symbols are followed by numeric or some mention not of much importance. Hence the presence of these symbols is a good indicator for the words before or after them for not to be a named entity.

5.2 Results and Discussion

Table 3 shows the results of the CRF model with 'l2sgd'(Stochastic Gradient Descent with L2 regularization term) algorithm for 100 iterations. The c2 value corresponds to the 'L2 regression' which is used to restrict our estimation of w*. Experiments using the algorithms 'ap'(Averaged Perceptron) and 'pa'(Passive Aggressive) yielded almost similar F1-scores of 0.96. Table 5 shows

Tag	Precision	Recall	F1-score
B-Loc	0.958	0.890	0.922
I-Loc	0.867	0.619	0.722
B-Org	0.802	0.600	0.687
I-Org	0.385	0.100	0.159
B-Per	0.908	0.832	0.869
I-Per	0.715	0.617	0.663
OTHER	0.974	0.992	0.983
weighted avg	0.963	0.966	0.964

Table 3: CRF Model with 'c2=0.1' and 'l2sgd' algo.

Tag	Precision	Recall	F1-score
B-Org	0.55	0.61	0.58
I-Per	0.43	0.50	0.47
B-Per	0.76	0.76	0.76
I-Loc	0.50	0.59	0.54
OTHER	0.98	0.97	0.97
B-Loc	0.83	0.84	0.84
I-Org	0.09	0.13	0.11
weighted avg	0.94	0.94	0.94

Table 4: Decision Tree Model with 'max-depth=32'

the weighted average feature specific results for the CRF model where the results are calculated excluding the 'OTHER' tag. Table 4 shows the results for the decision tree model. The maximum depth of the model is 32. The F1-score is 0.94. Figure 1 shows the results of a Decision tree with max depth = 32. Table 6 shows the weighted average feature specific results for the Decision tree model where the results are calculated excluding the 'OTHER' tag. In the experiments with BiLSTM we experimented with the optimizer, activation functions, no of units and no of epochs. After several experiment, the best result we came through was using 'softmax' as activation function, 'adam' as optimizer and 'categorical cross entropy' as our loss function. The table 7 shows the results of BiLSTM on our corpus using a dropout of 0.3, 15 epochs and random initialization of embedding vectors. The F1-score is 0.95. Figure 2 shows the BiLSTM model architecture.

Table 8 shows an example prediction by our CRF model. This is a good example which shows the areas in which the model suffers to learn. The model predicted the tag of '@Thirumalagiri' as 'B-Per' instead of 'B-Loc' because their are person names which are lexically similar to it. The tag of the word 'Telangana' is predicted as 'B-Loc' instead of 'B-Org' this is because 'Telangana' is a 'Location' in most of the examples and it is an 'Organization' in very few cases. We can

Feature	Precision	Recall	F1-score
Char N-Grams	0.73	0.56	0.62
Word N-Grams	0.88	0.59	0.70
Capitalization	0.15	0.02	0.03
Mentions, Hashtags	0.36	0.14	0.19
Numbers in String	0.01	0.01	0.01
Previous Word tag	0.78	0.19	0.15
Common Symbols	0.21	0.06	0.09

Table 5: Feature Specific Results for CRF

Feature	Precision	Recall	F1-score
Char N-Grams	0.42	0.72	0.51
Word N-Grams	0.57	0.59	0.58
Capitalization	0.19	0.31	0.23
Mentions, Hashtags	0.29	0.20	0.22
Numbers in String	0.06	0.16	0.07
Previous Word tag	0.14	0.20	0.16
Common Symbols	0.16	0.20	0.16

Table 6: Feature Specific Results for Decision tree

Tag	Precision	Recall	F1-score
BL	0.94	0.86	0.89
BO	0.76	0.56	0.64
BP	0.80	0.70	0.74
IL	0.41	0.55	0.47
IO	0.04	0.09	0.056
IP	0.33	0.52	0.40
OTHER	0.97	0.98	0.97

Table 7: Bi-LSTM model with optimizer = 'adam' and has a weighted f1-score of 0.95

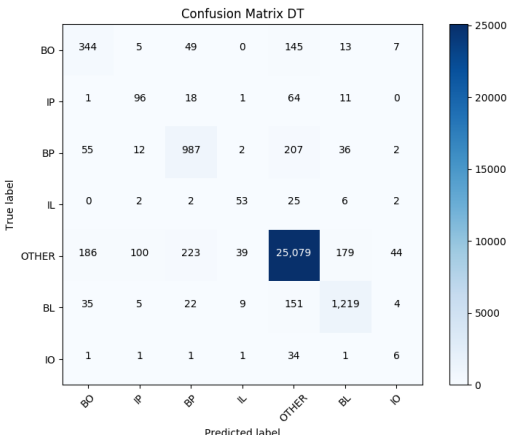

Figure 1: Results from a Decision Tree

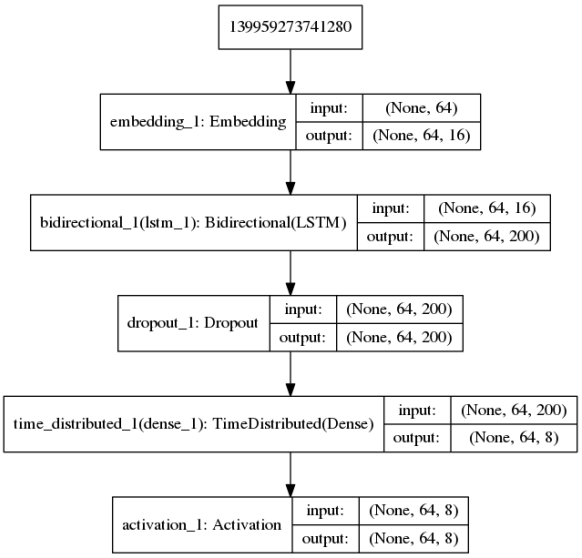

Figure 2: BiLSTM model architecture

also see '@MedayRajeev' is predicted as 'B-Org' instead of 'B-Per'. The model performs well for 'OTHER' and 'Location' tags. Lexically similar words having different tags and insufficient data makes it difficult for the model to train at times as a result of which we can see some incorrect predictions of tags.

6 Conclusion and future work

The following are our contributions in this paper.

1. Presented an annotated code-mixed Telugu-English corpus for named entity recognition which is to the best of our knowledge is the first corpus. The corpus will be made available online soon.

2. Experimented with the machine learning models Conditional Random Fields(CRF), Decision tree, BiLSTM on our corpus, the F1-score for which is 0.96, 0.94 and 0.95 respectively. Which is looking good considering the amount of research done in this new domain.

3. Introducing and addressing named entity recognition of Telugu-English code-mixed corpus as a research problem.

Word	Truth	Predicted
Today	OTHER	OTHER
paper	OTHER	OTHER
clippings	OTHER	OTHER
manam	B-Org	OTHER
vartha	I-Org	OTHER
@Thirumalagiri	B-Loc	B-Per
@Nagaram	B-Loc	B-Per
Telangana	B-Org	B-Loc
Jagruthi	I-Org	OTHER
Thungathurthy	B-Loc	B-Loc
Niyojakavargam	OTHER	OTHER
@MedayRajeev	B-Per	B-Org
@JagruthiFans	B-Org	B-Org

Table 8: An Example Prediction of our CRF Model

As part of the future work, the corpus can be enriched by also giving the respective POS tags for each token. The size of the corpus can be increased with more NE tags.The problem can be extended for NER identification in code-mixed text containing more than two languages from multilingual societies.

References

Andoni Azpeitia, Montse Cuadros, Seán Gaines, and German Rigau. 2014. Nerc-fr: supervised named entity recognition for french. In *International Conference on Text, Speech, and Dialogue*, pages 158–165. Springer.

Rupal Bhargava, Bapiraju Vamsi, and Yashvardhan Sharma. 2016. Named entity recognition for code mixing in indian languages using hybrid approach. *Facilities*, 23(10).

Eyamba G Bokamba. 1988. Code-mixing, language variation, and linguistic theory:: Evidence from bantu languages. *Lingua*, 76(1):21–62.

William B Cavnar, John M Trenkle, et al. 1994. N-gram-based text categorization. *Ann arbor mi*, 48113(2):161–175.

Pius von Däniken and Mark Cieliebak. 2017. Transfer learning and sentence level features for named entity recognition on tweets. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 166–171.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Kevin A Hallgren. 2012. Computing inter-rater reliability for observational data: an overview and tutorial. *Tutorials in quantitative methods for psychology*, 8(1):23.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Stephen Huffman. 1995. Acquaintance: Language-independent document categorization by n-grams. Technical report, DEPARTMENT OF DEFENSE FORT GEORGE G MEADE MD.

Faryal Jahangir, Waqas Anwar, Usama Ijaz Bajwa, and Xuan Wang. 2012. N-gram and gazetteer list based named entity recognition for urdu: A scarce resourced language. In *24th International Conference on Computational Linguistics*, page 95.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444.

P Majumder, M Mitra, and BB Chaudhuri. 2002. N-gram: a language independent approach to ir and nlp. In *International conference on universal knowledge and language*.

Kovida Nelakuditi, Divya Sai Jitta, and Radhika Mamidi. 2016. Part-of-speech tagging for code mixed english-telugu social media data. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 332–342. Springer.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Kamal Sarkar. 2015. A hidden markov model based system for entity extraction from social media english text at fire 2015. *arXiv preprint arXiv:1512.03950*.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018a. Language identification and named entity recognition in hinglish code mixed tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 52–58.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018b. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Jenny Linet Copara Zea, Jose Eduardo Ochoa Luna, Camilo Thorne, and Goran Glavaš. 2016. Spanish ner with word representations and conditional random fields. In *Proceedings of the Sixth Named Entity Workshop*, pages 34–40.

A Appendices

Category	Hash Tags
Politics	#jagan, #CBN, #pk, #ysjagan, #kcr
Sports	#kohli, #Dhoni, #IPL #srh
Social Events	#holi, #Baahubali #bathukamma,
Others	#hyderabad #Telangana #maheshbabu

Table 9: Hashtags used for tweet mining