Identifying Languages at the Word Level in Code-Mixed Indian Social Media Text

Submitted By:
Mohammad Ishan Rayeen (20162115)
Aneel Balda (20162035)
Syed Muzammil (20162119)

Submitted To:
Dr. Manish Shrivastava
Natural Language Processing
24th Nov. 2017

Abstract:

Language identification at the document level has been considered an almost solved problem in some application areas, but language detectors fail in the social media context due to phenomena such as utterance internal code-switching, lexical borrowings, and phonetic typing. all implying that language identification in social media has to be carried out at the word level. In this paper we reports a study to detect language boundaries at the word level in chat message corpora in mixed English-Telegu.

Dataset:

We have collected the dataset by parsing English-Telegu groups of their posts as well as comments and manually tokenizing and tagging them of the following 4 categories:

1. English as en.
2. Telegu as te.
3. Named entities as ne.
4. Others/universals as univ.

Sample dataset is as follows:

I	**en**
think	**en**
he	**en**
went	**en**
to	**en**
allu	**te**
of	**en**
Atharintiki	**ne**
Daredhi	**ne**
movie	**en**
;	**univ**

The dataset is tab separated with the actual word followed by the appropriate tag.

Model:

We have used multilayer perceptron model with 1 hidden layer and 66 internal nodes, firstly we converted each english word into a 64 dimension vector using Google's word2vec[1] and for the Telegu we have used a dummy vector of same dimension as that of the english vector.

We take trigram as input sequence by concatenating the feature vector of each trigram in sequence. So the input to our model is 3*vector dimension. The trigrams are taken so that MLP can learn the context of the words by learning their position in the training set. After training a forward pass is made for testing set and the class having highest probability corresponding to input word, will be allotted to that class.

After that we have added other various features such as:

1. If the word is in enchant[2] dictionary or not.
2. Case type of the word.
3. Length of the word.
4. If the word is alpha, alphanumeric, numeric, digit

Results:

	precision	recall	f1-score	support
eng	0.95	0.91	0.93	327
ne	0.79	0.57	0.67	94
te	0.79	0.95	0.86	208
univ	0.98	0.96	0.97	98
avg/total	0.89	0.89	0.88	727

References:

[1] Google's word2vec: https://code.google.com/archive/p/word2vec/

[2] Enchant Dictionary Python: http://pythonhosted.org/pyenchant/tutorial.html