Code-Borrowedness of English words in Hindi Language

Ram Mohan
R&D Department
Flytxt, India
ram.mohan@flytxt.com

Muhammad Arif
R&D Department
Flytxt, India
muhammad.arif@flytxt.com

Jobin Wilson
R&D Department
Flytxt, India
jobin.wilson@flytxt.com

1 INTRODUCTION

The goal of IKDD-CODS 2017 data challenge is to develop a metric to rank a set of candidate words according to the likeliness of code-borrowing from English to Hindi.

1.1 Evaluation Metric

To determine the ground truth a survey would be conducted where a user has to choose between Hindi statements with one having the target Hindi word and the other with corresponding English word . The words are ranked based on the sum('English word preferred') - sum('Hindi word preferred').

1.2 Data Provided

i. To come up with a metric we are shared with twitter data along with words tagged as Hindi(HI) or English(EN) or Named Entity(NE) or Other(OTH)
ii. Sample survey responses for a subset of 12 words are also shared.

2 METRIC PROPOSED

As can be seen in block diagrams 1 and 2 the metric proposed involves identifying the features and learning the weights for the features which are elaborated in the following section.

2.1 Pre-processing and Feature Extraction

2.1.1 Tag Tweets. Tweets were tagged as HI or EN or CMH or CME or CMEQ. For tagging the tweets only ENC(English count) and HIC(Hindi count) word counts were considered (ie, OTH and NE were not included for counting of total words for a tweet).

Table 1: Tweet Tagging Rules

Tweet Tag	Rule
EN	ENC/(ENC+HIC)>.9
HI	HIC/(ENC+HIC)>.9
CME	ENC/(ENC+HIC)>.5
CMH	HIC/(ENC+HIC)>.5
CMEQ	ENC/(ENC+HIC)==.5

2.1.2 Hash Tags. It was noticed that there were a lot of English tweets when compared to Hindi and CMH tweets, which was adding bias/noise of its own. To reduce it, we first identified the hash tags present in Hindi tweets, then we filtered only those tweets which were having these tags. It is intuitive that the survey for determining ground truth has equal prior for all the words, where as if there are too many English tweets which are irrelevant then they introduce a noise.

2.1.3 Stemming and feature statistics. It was identified that words such as 'film/films' needs to be correctly processed, hence stemming was performed which gave us a cleaner statistics for the keywords involved.

Natural Language Toolkit(NLTK) was used for pre-processing. After removing special characters, tweets were tokenized into words, converted to lowercase, filtered out stopwords and individual words were stemmed to its root form.

Feature list mentioned in the problem statement were used as is except that they were extracted from tweets based on Hindi Hash-Tag
$HHTU_{hi}$ - Is the number of users who have used the keyword in their Hindi tweets
$HHTU_{en}$ - Is the number of users who have used the keyword in their English tweets
$HHTU_{cmh}$ - Is the number of users who have used the keyword in their CMH tweets
$HHTU_{cme}$ - Is the number of users who have used the keyword in their CME tweets
$HHTU_{cmeq}$ - Is the number of users who have used the keyword in their CMEQ tweets
$HHTT_{hi}$ - Is the number of hindi tweets containing the keyword
$HHTT_{en}$ - Is the number of english tweets containing the keyword
$HHTT_{cmh}$ - Is the number of cmh tweets containing the keyword
$HHTT_{cme}$ - Is the number of cme tweets containing the keyword
$HHTT_{cmeq}$ - Is the number of cmeq tweets containing the keyword

2.2 Metric

Table 2: Spearman Correlation Coefficient for Algorithms

Algo	3 fold CV	12wor S
Ordinal Regression	-.03	0.97
Linear Regression	.67	.91
Non-Linear Regression	.73	.94
HandCrafted	.7915	.7915

Table 2 tabulates the different algorithms, their cross-validated Spearman Correlation Coefficients and their "12 data point" trained Spearman Correlation Coefficients. As there were very few data points available and we were not able to add data points our choice of model is Handcrafted weights model.

2.2.1 Handcrafted Final Metric. The metric used for ranking was
$HindiTaggedUserRatio(w) =$
$(HHTU_{hi} + HHTU_{cmh})/(HHTU_{en})$

,
....$
DOI:

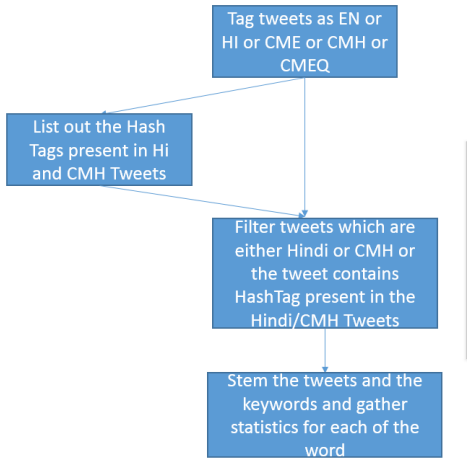

Figure 1: Preprocessing and Feature Extraction

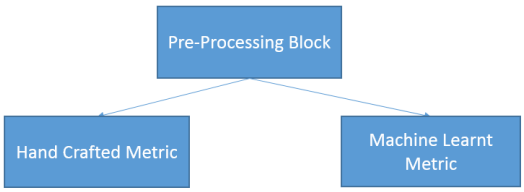

Figure 2: Learning weights of the Metric

$HindiTaggedTweetRatio(w) = (HHTT_{hi} + HHTT_{cmh})/(HHTT_{en})$

$FinalScore(w) = (HindiTaggedUserRatio(w) + HindiTaggedTweetRatio(w))/2$

2.2.2 Machine Learning based - For Discussions only. For any supervised Machine Learning algorithm to be successfull we require sufficient data, hence we tried to devise multiple ways to add more keywords with ranks but we were confident with the extra data added. But still we will list down the models that were tried and were found to be promising on the 12 samples that were available.

1. Ordinal Regresssion/Learning to Rank : As part of this the features were made to learn the ordering 2. Linear Regression: As part of this weights for features were learnt against the normalized ground truth as y. 3. Non-Linear Regression: Kernalized-Ridge and SVR were modelled with ground truth as the y. 4. Neural Network: A neural network was trained for the model $y = (w_1'X + a)/(w_2'X + b)$.

Python Libraries used :
scikit-learn for Linear Regression and Non-Linear Regression
mord for Ordinal Regression
keras for Neural Networks

3 CONCLUSIONS

In a limited data domain, The handcrafted metric along with proper preprocessing performed better. And the metric is $FinalScore(w) = (HindiTaggedUserRatio(w) + HindiTaggedTweetRatio(w))/2$