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 them being code-borrowed from English to Hindi, by analyzing social media conversations.

1.1 Ground Truth

The user preference towards usage of a Hindi word in its Hindi form as opposed to its English form in a Hindi sentence, is determined through a survey. From the survey responses, difference between the total number of instances wherein the word is preferred in its Hindi form and the instances wherein it is preferred in its English form is calculated to form the *ground truth metric*. The survey responses for 12 words are available from 58 participants, to measure the effectiveness of our proposed metric.

1.2 Dataset

A social dataset consisting of approximately 0.24 million twitter conversations is provided for conducting the experiments. Tweets are labelled at the word level using tags such as as Hindi(HI), English(EN), Named Entity(NE) and Other(OTH).

2 THE PROPOSED METRIC

As depicted in Figure1, calculation of the proposed metric involves identifying the relevant features and constructing a scoring function. The procedure is detailed in the subsequent sections.

2.1 Pre-processing and Feature Extraction

2.1.1 Labelling Tweets. Tweets were labelled as HI, EN, CMH, CME or CMEQ based on the rules described in Table 1. For labelling the tweets, only English word count(ENC) and Hindi word count(HIC) were considered. Words tagged as OTH and NE were not considered for calculating the total number of words in 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 Estimating Tweet Relevance. It was observed that English tweets were significantly more in the dataset as compared to Hindi and CMH tweets, adding a bias. To counter this effect, we identified the hash tags present in Hindi tweets and then selected only those tweets having these tags, for our analysis. Henceforth we refer to this subset of tweets as *relevant tweets*. The intuition behind our approach is that all English tweets may not be equaly relevant in determining the code-borrowedness of words.

2.1.3 Stemming and word statistics. It was observed that several variants of a root word exist in the tweets. To derive accurate word statistics, we used stemming. For instance, our pre-processing would transform words such as film, films, filme etc. to its root form 'film'.

Tweets were converted to lowercase, tokenized and stop words were removed. Natural Language Toolkit(NLTK) was used for stemming, to convert words to its root form.

The following features were extracted from the *relevant tweets*

- RTU_{hi} - The number of users who have used the root word in their Hindi tweets.
- RTU_{en} - The number of users who have used the root word in their English tweets.
- RTU_{cmh} - The number of users who have used the root word in their CMH tweets.
- RTU_{cme} - The number of users who have used the root word in their CME tweets.
- RTU_{cmeq} - The number of users who have used the root word in their CMEQ tweets.
- RTT_{hi} - The number of Hindi tweets containing the root word.
- RTT_{en} - The number of English tweets containing the root word.
- RTT_{cmh} - The number of CMH tweets containing the root word.
- RTT_{cme} - The number of CME tweets containing the root word.
- RTT_{cmeq} - The number of CMEQ tweets containing the root word.

2.2 Metrics

We experimented with five models for ranking words based on their their code-borrowedness from English to Hindi. All the models made use on the features described in the previous section.

Our basic intuition was that the code-borrowedness index of a word could be expressed as function of the features that we extracted from the *relevant tweets*. In our machine learning based models, we attempted to learn this function from the 12 examples provided. A performance comparison of these models in a 3-fold

,
....$
DOI:

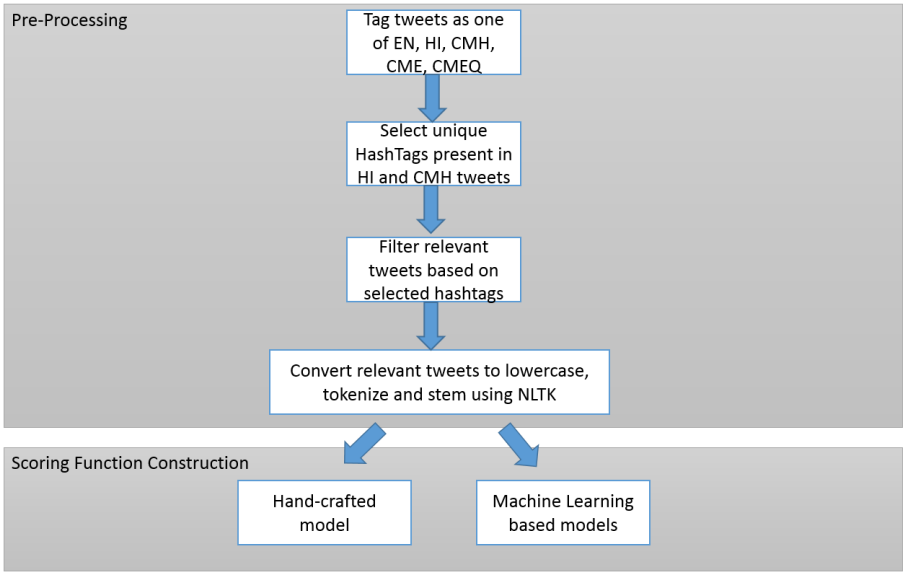

Figure 1: Model Pipeline

Table 2: Performance Comparison of the Proposed Models

Model	Spearman Correlation Coefficient
Ordinal Regression	-.03
Linear Regression	.67
Non-Linear Regression	.73
Neural-Network	.47
Hand-Crafted	**.7915**

cross-validation setting was done for the 12 words for which survey responses were available. Since the number of labelled examples were too few potentially causing poor generalization, we also experimented with a hand-crafted model. The evaluation metric used in all cases was Spearman Correlation Coefficient. Our results presented in the Table 2 indicate that the hand-crafted model gave superior results.

2.2.1 Hand-crafted Model. Our hand-crafted model calculated the code-borrowedness index of a word using the formulas described below.

$$RHTUR(w) = \frac{RTU_{hi} + RTU_{cmh}}{RTU_{en}} \quad (1)$$

$$RHTTR(w) = \frac{RTT_{hi} + RTT_{cmh}}{RTT_{en}} \quad (2)$$

$$BorrowednessScore(w) = \frac{RHTUR(w) + RHTTR(w)}{2} \quad (3)$$

2.2.2 Machine Learning based models. In our machine learning based approach, we used supervised learning algorithms for function approximation. The target variable was the *ground truth metric* and the input features were the ones described in Section2.1.3. Our machine learning models are listed below.

- Ordinal Regression: In this model, we attempted to learn a function to directly rank words
- Linear Regression: In this model, weights for input features were learnt and code-borrowedness score was modelled as a weighted linear combination of the input features.
- Neural-Networks: In this approach, code-borrowedness score was modelled as $y = \frac{w1'X+b1}{w2'X+b2}$.
- Non-Linear Regression: Kernalized-Ridge Regression and Support Vector Regression(SVR) models were used for function approximation.

We made use of scikit-learn[1] library for Linear Regression and Non-Linear Regression, mord[2] for Ordinal Regression and keras[3] for Neural Networks.

3 CONCLUSIONS

In this work, we evaluate several models to compute code-borrowedness index of words from English to Hindi based on twitter conversations. We observe that identifying the *relevant tweets* indicative of code-borrowedness of words is essential to derive features capable of ranking words appropriately in this context. Further, the limited availability of ground truth data inhibits machine learning approaches to effectively learn a function approximation capable of predicting code-borrowedness of words. Our proposed hand-crafted model deriving $BorrowednessScore(w)$ from simple features calculated from *relevant tweets* appear to be performing significantly better compared to the other models considered.

REFERENCES

[1] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.
[2] Fabian Pedregosa. mord: ordinal regression in python, 2016–. [Online; accessed 2017-02-03].
[3] François Chollet. Keras (2015). *URL http://keras. io.*