A Method to Estimate the Borrowedness Index of a word

R. PRASHANTH, Flytxt Mobile Solutions Pvt. Ltd.
DEEPAK. K, Flytxt Mobile Solutions Pvt. Ltd.
RAHUL VYAS, Flytxt Mobile Solutions Pvt. Ltd.
ASHUTOSH PATEL, Flytxt Mobile Solutions Pvt. Ltd.

Code Borrowing and Code mixing are two important phenomenon occuring during normal conversation of people. Estimating the amount of borrowness index can help in many aspects of multilingual information retrieval and natural language processing. In this work, we describe an approach to estimate the borrowedness index of word through social media analysis. The language tagged data was shared in the contest website, from which we extracted the actual tweet and carried out further processing.

Additional Key Words and Phrases: Code Mixing and Borrowing, Social Media Analytics, Multilingual data

ACM Reference format:
R. Prashanth, Deepak. K, Rahul Vyas, and Ashutosh Patel. 2016. A Method to Estimate the Borrowedness Index of a word. 1, 1, Article 1 (January 2016), 2 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Code Borrowing or Code Mixing is a linguistic phenomenon occurring during conversations. In these phenomenon, a word or a phrase from a foreign language is used as a part of the native vocabulary of the domain language. Estimation of code borrowing is important as it can help in many aspects of multilingual information retrieval and Natural language processing.

In this work, social media data was used for analysis. The data was preprocessed to remove unwanted text such as punctuation, stop words, non-ASCII text and duplicate comments. This is followed by creating bag-of-words model. Further processing steps are discussed in the next section.

2 MATERIALS AND METHODS

The detailed flow chart of the process is shown in the Figure 1 as shown below. Explanation of the steps is given in the subsections as given below.

2.1 Data Selection, Preprocessing, Bag-of-Words model and Term Frequency Matrix generation

The language tagged data that was shared in the contest website was taken and used for processing. This file contained the tweet id, and the language tags for each word in the tweet.

From this data, the number of English and Hindi tags were computed for each tweet, followed by selecting those tweets which are having at least one English and one Hindi tag. After this, the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2016 ACM. XXXX-XXXX/2016/1-ART1 $15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

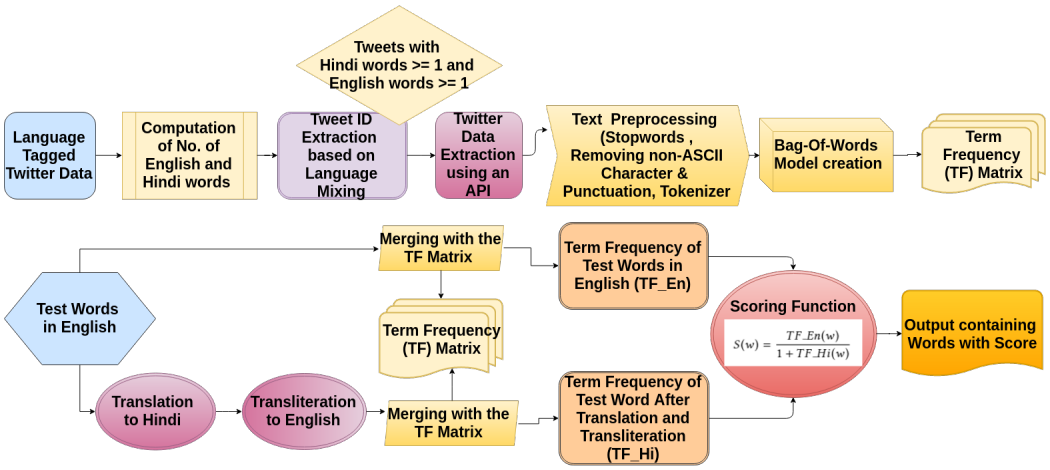

Fig. 1. Flow chart of the process

actual tweet data was extracted using the tweet ids and the tweepy API. Duplicate tweets were removed. The resulting data contained 133603 tweets.

After Twitter data extraction, the tweets were preprocessed for removing stop words, non-ASCII characters, punctuation and then the tweets were tokenised to create the Bag-of-Words model.

2.2 Test Words, Translation, Transliteration and Scoring

The number of occurrences of each word is computed and a Term Frequency (TF) matrix is generated.

The candidate test words are given as the input in the next step. The number of occurrences of the test words are computed by merging with the term frequency matrix. This gives us the term frequencies of test words in English, let's represent it as TF_En.

Along with this, the candidate test words are translated from English to Hindi using the "Translator" API (https://pypi.python.org/pypi/translate) and then transliterated back to English using the "Transliterator" API (http://silpa.readthedocs.io/projects/transliteration/en/latest/_modules/transliteration/core.html).For instance, for the word 'love', the translation to Hindi is 'ffh'and transliteration to English is 'pyaar.' Now the number of occurrence of these translated-transliterated test words are also computed by merging with the term frequency matrix. This gives us the the term frequencies of test words after translation and transliteration, let's represent it as TF_Hi.

Scoring for a word w is carried out using the following formula given below. The scores are such that higher scores indicate higher importance.

$$S(w) = \frac{TF\_En(w)}{1 + TF\_Hi(w)} \tag{1}$$

After the analysis, an csv file is generated with the words and corresponding rankings.