Romanagari Detection in Twitter

14 Oct 2015

Hrishikesh Terdalkar - Shubhangi Agarwal
Motivation

- Why Twitter?

- Most NLP techniques deal with English text only

- Tweets are often of the form:

  “Yeh kaisi field placings lagayi hain? Powerplay mein slip? Via @ARangarajan1972 #IndvsPak”
Romanagari = Noise
Goal

- Collect and create a quality tweet-dataset containing Romanagari words
- Romanagari Text Detection
- (possibly) Translate to English language

Languages Targeted

- Hindi
Steps

1. Create a dictionary of Romanagari words
2. Detect Romanagari text mixed with English text
3. Translate to English
Sounds easy?!
Challenges

1. Data Collection
   i. Search terms
   ii. Noise (different languages)
   iii. Disambiguation (polysemy in Hindi and English)

2. Detect and differentiate between English and Romanagari text
   i. Phonetic typing
   ii. SMS language
   iii. Spelling errors
   iv. Disambiguation
Challenges

3. Handle commonly occurring inflections in the social media text
   i. whatttttt!, whennnn??, kyunnnn??
   ii. mann, bool, bol

4. And many more (yet to be encountered)
Approach

1. Data
   i. Collection
      ➢ Frequent Romanagari words
      ➢ Tweepy
      ➢ SMS language
   ii. Synthetic Generation

2. Language detection/correction
   i. Tools available (PyEnchant, langid, langdetect, guess-language etc)

3. Almost phonetic representations
   i. Metaphone
   ii. Double Metaphone
   iii. Soundex

➢ Also used for Romanagari text detection
Strategies

- Find frequently used Romanagari words in tweets/social media. (Different from “most frequent” Hindi words from other corpora such as books / wiki)

- Try to obtain annotated-datasets from social media such as facebook from existing papers and frequency analysis on this smaller “spoken-hindi” dataset.

- Context analysis (if possible)
  - n-grams
So far..

➔ Python
➔ Twitter collection
  ◆ most frequent hindi words as FILTER
  ◆ low success rate on tweets + lot of noise
  ◆ explored synthetic generation\[^3\]
➔ Exploration of existing classifiers
  ◆ *PyEnchant*: a spellchecking library for Python based on Enchant
  ◆ *langdetect*: python implementation of “language-detection” Java library
  ◆ *langid*: language identification, n-gram, 97 languages, scores for multiple languages
➔ Soundex / Metaphone Experiments
Soundex vs Double Metaphone

"kyun"

```python
>>> s("kyun")
'K50000'
>>> s("kyunn")
'K50000'
>>> s("kyunnnnn")
'K50000'

>>> doublemetaphone("kyun")
('KN', '')
>>> doublemetaphone("kyunn")
('KN', '')
>>> doublemetaphone("kyunnnnn")
('KNNN', '')
```

"haan"

```python
>>> s("haan")
'H50000'
>>> s("haaannn")
'H50000'

>>> doublemetaphone("haan")
('HN', '')
>>> doublemetaphone("haaannn")
('HNN', '')
```

"what"

```python
>>> s("what")
'W30000'
>>> s("wwhaaattt")
'W30000'

>>> doublemetaphone("what")
('AT', '')
>>> doublemetaphone("wwhaaattt")
('TT', '')
```
Soundex vs Double Metaphone

"burp"

```python
>>> s("burp")
'B61000'
>>> s("burrrippp")
'B61000'
>>> doublemetaphone("burp")
('PRP', '')
>>> doublemetaphone("burrrippp")
('PRRPP', '')
```

"lol"

```python
>>> s("lol")
'L40000'
>>> s("lollll")
'L40000'
>>> doublemetaphone("lol")
('LL', '')
>>> doublemetaphone("lollll")
('LLL', '')
```

"boom"

```python
>>> s("booooooooom")
'B50000'
>>> s("boom")
'B50000'
>>> doublemetaphone("boooooooom")
('PM', '')
>>> doublemetaphone("boom")
('PM', '')
```

"ah / oh"

```python
>>> s("ah")
'A00000'
>>> s("aaahhh")
'A00000'
>>> doublemetaphone("ah")
('A', '')
>>> doublemetaphone("aaahhh")
('A', '')
```
Tweet Collection
Plan

➔ Better dataset collection strategies
➔ Better synthetic generation than mentioned in [3]
➔ Perform experiments to test feasibility of Soundex/Metaphone for Hindi
➔ Pre-processing tweets followed by language identifiers with modification
➔ Compose a list of Hindi-specific disambiguation rules
➔ Detect Romanagari words
➔ Annotate / Attach English meaning to Romanagari words
References

6. Proceedings of Social india 2014
7. Tweepy: [https://github.com/tweepy/tweepy](https://github.com/tweepy/tweepy)
Questions?
Thank You