Language Detection in Code-Mixed Social Media Text

Alefiya Naseem
Paarth Kotak
Siddhesh Acharekar

What is code mixed data?

The freedom one can afford from social media platforms often leads it's users to deviate from what is the accepted norm of communication in language.

One of these deviations happens when users switch between languages.

Code-mixed text adopts the vocabulary and grammar of two or more languages and often forms new structures based on its users.

For example, here's how somebody could write 'Snell library is very good, don't you agree?' in code-mixed English-Hindi:

Snell library bohot accha hain, don't you agree? :)

Project Task

Our goal with this project is word level language identification for a sentence as seen in the previous slide.

For the given example, the end result will be:

Snell//EN library//EN bohot//HI accha//HI hain//HI, don’t//EN you//EN agree//EN ?//UN :)//EMT

Where each word is assigned a language tag from either of EN (English), HI (Hindi), UN (Universal), EMT (Emoticon).

Data

- Considering the claim that code-mixing is most frequent among young, multilingual speakers, we decided to use an Indian student community in the 20-30 age group as our data source.
- To save data collection and annotation time we have taken data with permission from previous researchers on this topic who scraped a Facebook group and 11 of its users posts and comments.
- Posts having words in any language other than Hindi or English were discarded, leaving us with 12936 code-mixed posts having 201412 words in total.
- Each language tag associated with each word was one of ‘EN’(English), ‘HI’(Hindi), ‘UN’(Universal), or ‘EMT’(Emoticon).

Previous Work

The problem of language identification has been investigated for half a century (Gold, 1967) and that of computational analysis of code switching for several decades (Joshi, 1982), but there has been less work on automatic language identification for multilingual code-mixed texts.

Previous Work

- U Barman, A Das and J Wagner worked on this problem with English-Bengali code-mixed text which is our baseline.
- Their approach began with an unsupervised dictionary based method where the training data itself contributed to a dictionary leading to a train accuracy of 93.64%.
- Under supervised learning approaches with char-n-grams, presence in dictionaries, length of words and capitalizations as features and SVM and CRF's as models, they received 95.21% and 95.76% accuracies.
- Since Bengali was a language none of us knew, we chose to work on English-Hindi code mixed text.

Building on previous work

- Changed dictionaries to avoid overhead and work with more social-media specific dictionary.
- Tested with more Supervised Learning Algorithms in addition to SVM and CRF namely Naive Bayes, and Random Forests.
- Added more features in addition to the ones already tried by the reference paper.
- Implemented an LSTM network.

Methodology

- Word-Level Classification without Contextual Clues
  - Char n-grams
  - Length of word
  - Capitalization
  - Number of capitalizations
  - Suffixes
  - POS tags
  - Presence of digits
- Word-Level Classification with Contextual Clues
  - SVM with added feature: previous word and next word
  - Conditional Random Fields with added features: previous word, next word, previous POS tag and next POS tag.
- Neural Learning with LSTM

Preprocessing

- Since our approach takes a contextual and non contextual look at the problem we have 2 different preprocessing steps for each.
- For the non contextual way we combined our entire corpus into one and created a list of words and a list of its corresponding language tags.

```
In [93]: just_words
         executed in 22ms, finished 21:35:2
Out[93]: ['Han',
          'wo',
          'bhi',
          'baat',
          'hai',
          'its',
          'not',
          '#',
          'unionbudget2015',
          'its',
          '#',
          'buredinkabudget',
          'khana',
          ',',
          'cmputer',
          ',',
          'kapde',
          ',',
          'ghar',
```

```
In [94]: just_tags
         executed in 17ms, fi
Out[94]: ['HI',
          'HI',
          'HI',
          'HI',
          'HI',
          'EN',
          'EN',
          'UN',
          'EN',
          'EN',
          'UN',
          'EN',
          'HI',
          'UN',
          'UN',
          'UN',
          'HI',
          'UN',
          'HI',
```

Preprocessing

- For the contextual way we retained sentence structure and appended the language tags to each word only separated by '\\'.

In [96]:
```
js_as_json[1]['lang_tagged_text']
```

executed in 5ms, finished 23:30:22 2018-11-25

Out[96]:
```
'its\\EN not\\EN #\\UN unionbudget2015\\EN its\\EN #\\UN buredinkabudget\\EN khana\\HI ,\\UN cmputer\\UN ,\\UN kapde\\HI ,\\UN
ghar\\HI sab\\HI mehenga\\HI .\\UN bewakuf\\HI bnaya\\HI sirf\\HI inhone\\HI '
```

Features used

- Char n-grams(n=2 to 5): almost always highly correlated to the language (Cavnar and Trenkle,1994)
- Length: raw length of a word oppose to the technique used in the previous work (avg EN word > avg HI word by at least 1 alphabet)
- Capitalization: positional capitalization that emphasizes on the importance
- Number of capitalization: how many letters in the word are capitalized
- Suffixes: the affix placed after the stem of the word can also be indicative of a language
- POS tags: assigned part of speech tag of each word
- Presence of digits: if the word consists of any digit

Features rejected

- CountVectorizer / Tf-Idf.
- Soundex or phonetic matching.
- Readability scores.
- Maximum repetition of unigrams and bigrams.
- Presence in dictionaries.

Methods - Non-contextual

- Dictionary-based (Wordnet and Twitter scraped)
- Bayesian classifiers are used to classify a document (word) D to one of a set of predefined categories (languages) C = {HI, EN, EMT, UN}. We make the assumption that each feature is conditionally independent of other features given the language.
- SVM is implemented using a polykernel classifier which is quite popular in natural language processing tasks.
- Random Forests language models have the potential to generalize well to unseen data, even when a complicated history is used (Peng Xu and Frederick Jelinek)

Methods - Contextual

- Conditional Random Fields
  - We use linear (chain) CRFs
  - It takes history into account in predicting optimal sequence of labels.
  - CRFs are conditional models, that is, they represent conditional probability distributions of the form p(y|x) between the inputs x and the outputs y.
  - Correlations do not matter as we take conditional distribution instead of joint distribution.
- SVM with Context
  - To obtain contextual information, we add the previous and next words as features to our existing set of features.
  - Polykernel SVM model is again used for classification.
- LSTM
  - Expected to perform well for language detection since language text often has long term dependencies. Hence, effective for tagging sequential data.

Experiments and Evaluation metric

Data: 12936 Facebook posts of 201412 words and tags.

Train, test: All experiments were carried out with a 75% - 25% train-test split.

Evaluation: Classification accuracy.

Baselines

- U Barman, A Das and J Wagner's work on English-Bengali code-mixed data is our main reference paper which we followed but decided to change their features since our corpus was different.
- Their experiments on English-Bengali text achieved the following accuracy results for these methods:
    - Dictionary: 93.64%
    - SVM: 95.21%
    - CRF: 95.76%

Results: Unsupervised Dictionary based model.

Dictionary-based Models	Accuracy
Word net	52.33%
Twitter based	56.66%

As expected, the dictionary-based approaches did not perform too well. However, the Twitter based dictionary performed marginally better since the words in it's corpus came from the same domain (Social Media) as that of the train and test data.

Results: Supervised Learning (Non-contextual features)

RandomForest (batch_size = 100, max_depth = 0) and SVM (C=1.0, L=0.001)

Model	Accuracy
Naive Bayes	87.36%
SVM	86.05%
Random Forest	88.87%

Results: Supervised Learning (Contextual features)

SVM (C=1.0, L=0.001) and CRF (algorithm = ‘lbfgs’, c1: 0.0183, c2: 0.0256)

Model	Accuracy
CRF	89.05%
SVM	87.32%
LSTM	84.61%