Sentiment Analysis on Hindi-English Code Mixed Data

Guide: Radhika Mamidi

Vrishank Shete (201305642)

Lokesh Mittal(201305650)

Gagandeep Chhabra (201305658)

- Introduction: Sentiment Analysis and Code Mixing
- Task Description
- Resources Developed
- Methods - Feature Engineering
- Results

Introduction

- Growing availability and popularity of opinion-rich resources such as online review sites, personal blogs, microblogging websites like twitter, facebook and other social networks new opportunities and challenges arise.

- The Indian pages on these websites often contains hin-eng mixed comments, and mostly the users writes these comments in Roman script due to difficulty in writing Devanagari.

- Marketers can use this to research public opinion of their company and products, or to analyze customer satisfaction.(can know what's right and what went wrong in other words positive and negative aspects of there service/product)

Task Description

Automatically extracting sentiment from a given **text sample** is the task at hand. One can observe code-mixing in user generated content on social media, especially from multilingual users (people knowing more than one language). Such data neither has any specific spelling standards nor any formal grammar rules.

We take up a training dataset which contains comments of labelled with sentiment classes, described as follows:

- **-1 – Negative**
- **0 – Neutral**
- **1 – Positive**

Task is to, use the training set to construct a model which then can be used to predict the class labels of the test dataset.

Resources Developed

- **Dataset:** Crawled raw data from Facebook's famous public pages like Narendra Modi, AAP, Garbage Bin etc.
  Cleaned, refined and labelled data with total count of 6700.
- **Dictionaries**: Using Hindi wordnet dictionary by IITB and Hindi dictionary developed at IIITH, developed two new dictionaries by following method:
  - First converted the Devanagari words to WX using IIITH Shallow parser.
  - Using reverse-WX notation rules, converted words to roman script.

  For example:

Original word(in Devnagri)	Intermediate word (WX Notation)	Words placed in our dictionary
अच्छा	acCA	acchaa, accha, acha
ग़ुलाम	gZulAma	gulaama, gulama, gulaam, gulam

Technique & Feature Engineering

- **Support Vector Machine (SVM)**

  a. Pre-processing of comments

  POS tagging, lowercase, handling negation, Normalization(de-vowelization and permutations of words)

  b. Feature Vector Creation

  word-n-gram, lexicon score based features (prior polarity using dictionary), POS-tags counts separate for english and hindi words, emoticons, PMI based scoring.

Method-1

Feature Vector used: Unigrams and bigrams without normalization

Accuracy obtained: 49% (approx.)

ML Tool used: LibSVM

Method-2

Feature Vector used: Unigrams and bigrams with normalization

Accuracy obtained: 53% (approx.)

ML Tool used: LibSVM

Method-3

Feature Vector used: Unigrams and bigrams with normalization, lexicon score based features for both english and hindi, emoticons

Accuracy obtained: 57% (approx.)

ML Tool used: LibSVM

Method-4

Feature Vector used: Unigrams and bigrams with normalization, lexicon score based features for both english and hindi, emoticons and POS Tags based feature for both english and hindi

Accuracy obtained: 59% (approx.)

ML Tool used: LibSVM

Method-5

Feature Vector used: Unigrams and bigrams with normalization, lexicon score based features for both english and hindi, emoticons and POS Tags based feature for both english and hindi , PMI score with scaling

Accuracy obtained: 61% (approx)

ML Tool used: LibSVM

Final 10-fold cross validation accuracy: 60% (approx.)

Tools/Resources Used

- IIIT-H's Shallow Parser
- LibSVM and LibLinear for SVM Classifier
- CMU’s ARK-POS Tagger
- English WordNet Dictionary

Thank You