Peer-to-Peer Insult Detection in Online Communities

Priya Goyal (10535)
prigoyal@iitk.ac.in
Dept. of Mathematics and Statistics

Gaganpreet Singh (10258)
gpskalra@iitk.ac.in
Dept. of Computer Science and Engineering

Guide: Prof. Amitabha Mukerjee
amit@cse.iitk.ac.in
Dept. of Computer Science and Engineering

Indian Institute of Technology, Kanpur India

Problem Statement

- Detecting comments intended to be insulting to other participant in blog/forum conversation.
- Comments intended to be insulting towards non-participants are not labeled as insults.
- Insults include: Crude language, Taunts, Slurs, Racism, Disguise, Unrefined language.

Motivation

- Insulting comments hurt user feeling.
- Discourage user participation and prevent new comers from participating.
- Frustration while searching for searching for specific information on some site.
- Large amount of increasing data difficult to be moderated by a human moderator manually.

Related Work

- Most work involves static dictionary use to look for offensive words like reference to handicap .
- Parts-of-speech (POS) n-grams, pattern matching approaches like "get lost" have been extensively studied.
- Due to flexibility of conversation, these approaches are rigid and lack generality.

Challenges Involved

- Grammatical mistakes: "What on earth a BIGGOT like you is doing walking on the face of earth?"
- Typography: s h i t (shit)
- People circumvent dictionary: @$$hole (asshole)
- Wordplays: kucf oyu
- Insult of non-participant-> not an insult
- Sarcasm: "Sometimes I don't know whether to laugh at you or pity you."
- Innuendo e.g. "Only cowards, thieves, cheats and liars hide behind pseudonyms."

Methodology

STEP 1 • Normalization

STEP 2 • Feature Extraction (vector model)

STEP 3 • Feature Selection

STEP 4 • Classification

Normalization

- Remove unwanted Strings: \\xc2, \\n, html tags
- Stemming: 'retarded' -> 'retard'
- Intended form: 'ur' -> 'you are'
  'nopes' -> 'no'
  'sh#t' -> 'shit'
  '@$$hole'-> 'asshole'

Feature Extraction

- Text strings converted to vector
- Bag-of-Words representation
- Tokenization: Tokens can be 'word' or 'n-gram'
- Counting: count of each token is a feature.
- Normalizing using Tf-idf score

Additional Features

- **Skip Grams**: Pair of long-distance words e.g. "you must be an idiot" -> you-idiot
- **Second-person' feature**: Words following 'you are', 'you'

Features Selection

- Best feature selection using '**Chi-Square**' test. This test is used to find if a pair of categorical variables on a sample are independent
- Features with maximum chi-square statistics w.r.t labels are selected.
- Categorical variables: insult/ non-insult token present or not

Classification

- Two machine learning algorithms Logistic Regression, SVM (with different kernels) used to learn a model on generated feature vectors.
- Logistic regression + SVM combined give better results than others.

Implementation and Results

- Accuracy without applying our hypothesis: 85.2381
- Accuracy with Skip Grams (2 words skipped) included: 86.5079
- Accuracy with Second-person rule included: 86.7725
- Accuracy with both Skip Grams and Second-person rule included in table:

Skips	Accuracy	Precision	Recall
3 skips	86.8481	0.7658	0.7172
2 skips	86.8481	0.7674	0.7143
2 , 3 skips	86.9237	0.7698	0.7143
2, 3, 4 skips	86.6591	0.7599	0.7172

Future Work

- False Positives: "you republican politicians will never get it in your head
- New Features like user id, comment thread length, replies to a comment etc.
- Sarcasm and Innuendo.

References

Author/ Year	Work
Ellen Spertus, 1997	Dictionary, Pattern Matching
Altaf Mahmud, Kazi Zubair Ahmed, Mumit Khan, 2008	Rules to extract semantic information to detect insults.
Amir H. Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin, 2010	Machine learning approach to multi – level classification using abusing and insulting language dictionary .
Carolyn P. Rose, Guang Xiang, Jason Hong, 2012	Topical feature (using LDA) and Lexical feature building and use of Machine learning algorithms.

- For Badwords file: http://urbanoalvarez.es/blog/2008/04/04/bad-words-list/
- For starter code: http://www.kaggle.com/c/detecting-insults%25E2%2580%2593in-social-commentary/forums
- For dataset: http://www.kaggle.com/c/detecting-insults-in-social-commentary/data