Peer-to-Peer Insult Detection in Online Communities

4th April, 2013

By:
Priya Goyal (10535)
Gaganpreet Singh Kalra (10258)
IIT Kanpur

Advisor:
Prof. Amitabha Mukerjee
IIT Kanpur

Problem Statement

- Detecting comments intended to be insulting to other participant in blog/forum conversation.

- Insults – profanity, racial slurs, other offensive language.

- Comments insulting to a non-participant are not labeled as insults.

Motivation

- Negative content– hurt user's feelings, barrier to the users/new comers participation.
- Frustration to users searching for information on sites.
- Large amount of increasing data difficult to be moderated by a human moderator manually.

Previous Work

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.

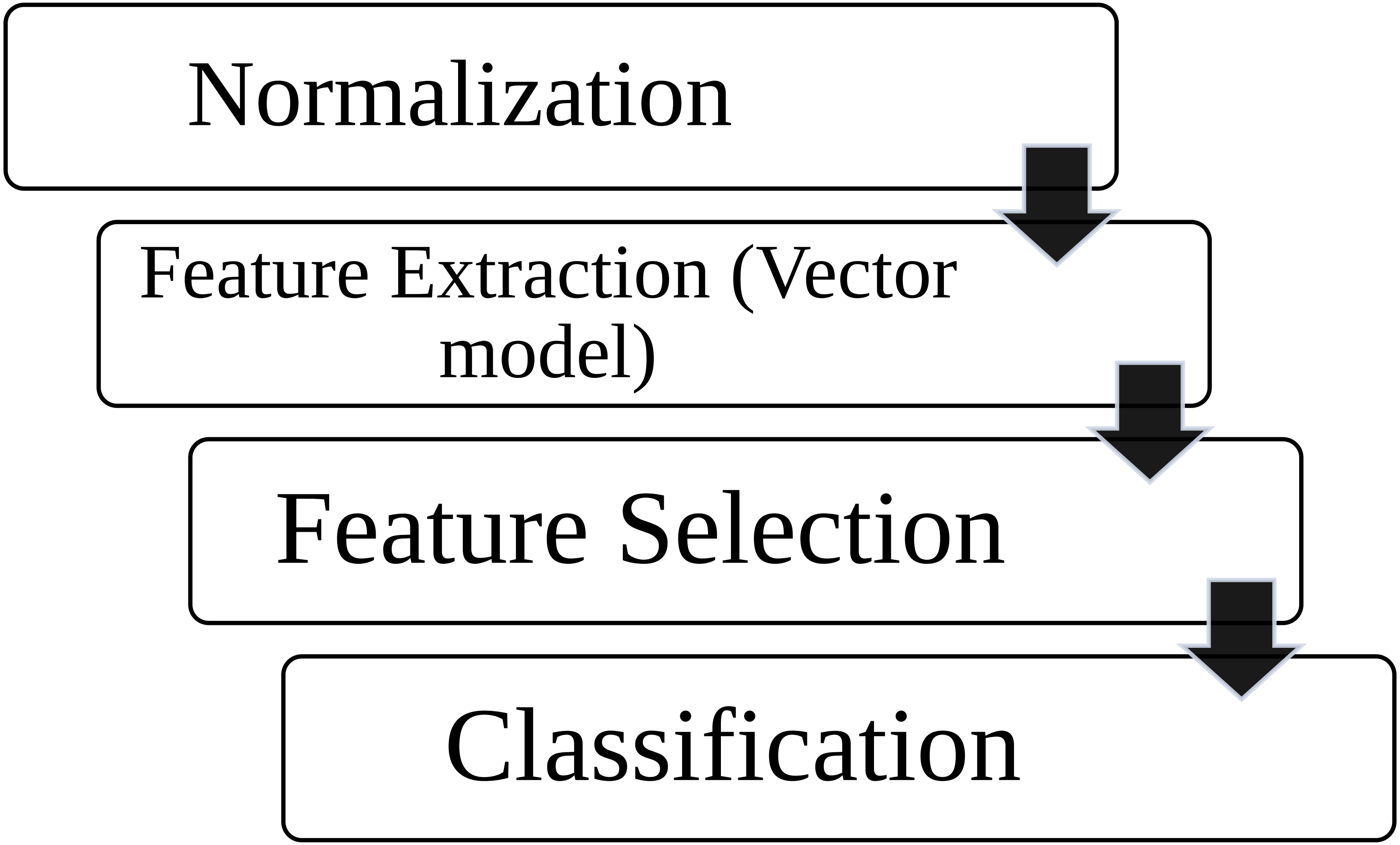
Previous Work

- Most works involve static dictionary and rules (pattern) matching approaches which are rigid and lack generality.
- Comments insulting towards a non-participant have also been considered as insults in these.

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



Normalization

Feature Extraction (Vector model)

Feature Selection

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 string converted to vector
- Bag-of-Words representation
  - Tokenization: Tokens can be ‘word’ or ‘ngram’
  - 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'
  - 40% of insults in our dataset had 'you', 'your' etc.

Feature 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.
- These are categorical variables which takes two values each:

  insult/ non-insult
  token present or not

Classification

- Two machine learning algorithms Logistic Regression, SVM (with different kernels)are used to learn a model on generated feature vectors.

- Logistic regression gives better results than others.

Results

- Accuracy without applying our hypothesis: 74.58%
- Accuracy with Skip Grams (2 words skipped) included: 74.63%
- Accuracy with Second-person rule included: 74.92%
- Accuracy with both Skip Grams and Second-person rule included: 75.13%

References

- Ellen Spertus. 1997. Smokey: *Automatic recognition of hostile messages*. In Proceedings of the Ninth Conference on Innovative Applications of Artificial Intelligence, pages 1058–1065.
- Altaf Mahmud, Kazi Zubair Ahmed, and Mumit Khan, 2008. *Detecting flames and insults in text*. In Proceedings of the Sixth International Conference on Natural Language Processing.
- Amir H. Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010 *Offensive language detection using multi-level classification*. In Proceedings of the 23rd Canadian Conference on Artificial Intelligence, pages 16–27.
- Xiang, G., Hong, J., & Rosé, C. P. (2012). *Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus*, Proceedings of The 21st ACM Conference on Information and Knowledge Management, Sheraton, Maui Hawaii, Oct. 29- Nov. 2, 2012.
- For badwords file: http://urbanoalvarez.es/blog/2008/04/04/bad-words-list/
- For starter code: www.kaggle.com/c/detecting-insults-in-social-commentary/forums
- For dataset: www.kaggle.com/c/detecting-insults-in-social-commentary/data

Thank You!

☺

Questions?

Normalization using Tf-idf score

- Tf-idf: Term frequency * inverse document frequency

$$\frac{\text{number of times a token occurs in a particular text string}}{\text{string/ fraction of documents in which the token occurs}}$$

- Number of occurrences not a good feature e.g. ‘the’ occurs in almost all the text strings.

Chi-Square Test

		Voting Preference		
	Congress	**BJP**	**SP**	**Total**
Male	200	150	50	400
Female	250	300	50	600
Total	450	450	100	1000

This test is used to find if a pair of categorical variables on a sample are independent

Chi-Square Test

- Say in a population, you can divide the members into 2 groups: Male and female (1st categorical variable takes two values)
- We can also divide the population on the basis of party they prefer: BJP, Congress, SP (2nd categorical variable takes 3 values)
- If the 2 variables are independent, the expected value E(Male, BJP) of # people who are male and prefer BJP = N(Male) * N(BJP)/(Total), similarly for other 5 combinations.
- We calculate $X^2 = \sum \frac{(Observerd(i,j) - Expected(i,j)))^2}{Expected(i,j)}$ where,

  i= {male, female}
  j= {BJP, Congress, SP}
- This is a measure of dependency in the 2 variables: higher values-> dependent and lower values-> independent