Noise reduction and normalization of microblogging messages

Doctoral Program in Informatics Engineering

Gustavo Laboreiro

Universidade do Porto

May 21st, 2018
Overview

1. Introduction to User-Generated Content
2. Writing style
3. Tokenization
4. Deobfuscation
5. Bots
6. Nationality
7. In closing
Introduction to User-Generated Content
What is User-Generated Content?

- Content created by regular folks
- On multiple platforms
- Share a number of properties
  - Self-focused
  - Short and to the point
  - Ubiquitous
  - …
  - Noisy
Messages can be noisy

Examples

- @syrinpt and I find slash fun in an silly way. but they’re pushing it seems like they want to transform all male friendships into slash
- Check it out! General_dreambox_manual Just got posted: http://trim.su/3NTg (by Fisicalsharing)
- @j_maltez Mother so young!?!!?!! :X
- January 25th 2010 ---- Solidarity game to help the people in HAITI ----- Estádio da LUZ ----- Benfica Foundation!
- Earthquake of 6,1 felt today in Haiti.
- :D #bgot RT @brpellican Dool get ready for Service Packs and constant reboots! RT @bitpop: (...) @BillGates is in Twitter
Microblogs can be useful

- Social approval or disapproval
- Prediction of riots
- Deriving sociologically relevant demographics
- Uncovering mental health of the population
- Real-time detection of earthquakes

XKCD by Randall Munroe
Our objective

Hypothesis
Can the noise in microblogs be addressed through learning methods and classification approaches?

Solving problems dealing with several forms of noise

In messages:
- Writing style
- Tokenization
- Profanity

In the population:
- Bots
- Nationality
Writing style
I was hacked!!!

Britney Spears

- I give myself to Lucifer every day for it to arrive as quickly as possible. Glory to Satan!

Fox News

- Breaking: Bill O Riley is gay

Willow Smith

- So Chris Brown is going to prison now breaking a window at ABC, but he didn’t go for hurting Rihanna? http://bit.ly/i6luln #karma
Social media is relevant

- Messages are used in court as evidence
  - Murder trials
  - Divorce cases
- Grounds to discharge from employment
- Shapes a public image
Our objective

To be able to attribute authorship of a message
- Did the supposed author write this?
- Who wrote it? (from a number of suspects)

But...
- The messages are short
- The languages are varied and often mixed
- The vocabulary is different
- There are many misspellings
- Abbreviations are common and inconsistent
User 1
- Tribute was 2 Damn short
- @ledisi kno she don’t need long 2 tear up a song my favorite Teena Marie song! Done well @BETAnwards
- Past 2 present. It’s your future. U gotta go through it 2 get 2 it. Good night/good morning. Live laugh love

User 2
- I love boys that have a nice smell. No perfume. Just nothing, but still nice. Lmao. Who gets me?
- @JelenaHolic lol no. I was about to lmao.
- @JelenaHolic lmao, im reading 13 reasons why, Selena’s new film! So good! Omg! :o
Groups of features

1. Quantitative Markers (number of words, average word length, number of hashtags, ...)
2. Marks of Emotion ("hahaha", "lol", "kkk", smileys, ...)
3. Punctuation (is it absent? Is it used at the end? What is being used? ...)
4. Abbreviations (what apparent abbreviations can we find?)
The experiments

- 120 users from Portugal, divided into groups of 3
- 2,000 distinct messages per user (no retweets)
- Datasets of size 75, 250, 1,250 and 2,000
- SVM classifier
- 5-fold cross validation
- Baseline: pure chance (1/3)
Comparing the impact of the size of the dataset

![Dataset size impact graph]

Gustavo Laboreiro (UP)  
Noise reduction and normalization of microblogging messages

2018-05-21  15 / 44
Conclusions

We learned that:

- It is possible to assign authorship to microblog messages
- It can be done based solely on stylistic features
- Marks of emotion was the most revealing feature group
- The “noise” is not random

With our implementation, for 3 suspects:

- We can be correct most of the time, based in only a small sample of messages
- We can be correct 2/3 of the time, based in larger samples
Tokenization
About tokenization

- Separate a text into atomic units (words, numbers, punctuation, URLs, smileys, . . .)
- Tokenization quality has an impact in later text processing stages (error correction, normalization, semantic analysis, . . .)
Message noise impairs the tokenization process

Examples

- “well I’m going to watch what I think will be a great ball game R.Madrid-Lyon... see you soon dear twitterers...”
- “(...) a great ball game R. Madrid - Lyon ... (...)
- “@carolisklassen Lets go and have a coca-cola and sleep on the table.”

Examples

- “@martiiinha don’t you have an account at last.fm?”
- “@Neiazitahh is going?? O.O That is serious!! xD”
- @VillegasBeliebr hey Taylor! Thanks :) ¡3
- (loOoL) olo (THE DRA.MA OF THAT DUMMYy is he won’tT leave HOME ALONEe.) Drama. Drama. LtsS of DRA,MA.
Our approach

- We can’t write rules to handle this, let alone maintain them
- Can we use a classification approach to solve this problem?
- Should a separating space character be inserted?
  - Next to a non-alphanum char
  - Unless a space char is adjacent

Location of decision points

Features used

About the character:

- **Character nature** (alphabetic, numeric, symbol, space, . . .)
- **Type of letter** (upper/lower case, accented, non-accented, vowel, . . .)
- **Type of symbol** (bar, dash, monetary, opening symbol, arithmetic, smiley nose, . . .)
- **Literal character**

Feature window:

- 10 characters on either side of the decision point
Our experiments

Two testing scenarios:

- Sall: Remove spaces from all decision points
- Sone: Remove just one space at a decision point (multiple tests per message)

Experimental set-up:

- 2500 messages tokenized manually
- SVM classifier
- 5-fold cross validation
- Compare with simple regex rules
- Baseline is always inserting the space
Our results

![Graph showing the performance of different methods with varying numbers of messages used for training. The graph compares the methods 'S all - SVM', 'S all - Rules', 'S all - Baseline', 'S one - SVM', 'S one - Rules', and 'S one - Baseline'. The y-axis represents the performance metric, and the x-axis represents the number of messages used for training.]
What we learned

- Text tokenization can be expressed as a classification problem
- Learning approaches can easily outperform rule-based methods
- A small feature window size is preferable to a large one (4–5)
- Improving the results is simple and easy, compared to rules
- “.” was the most problematic character (“-” came second)
Deobfuscation
Why?

Cursing:
- Taboo breaking for the sake of taboo breaking
- Expressing emotions
- Moral harm

Obfuscation:
- To blunt the impact
- To circumvent filtering
Why are we looking at swearing?

- It is more than error correction
- Ensure we are doing precision work in filtering (high recall)
- Identify toxic, hateful or abusive users or trolls
- Extract opinions/views from messages
Introducing the SAPO Desporto dataset

- Filtered by SAPO
- 2500 messages annotated
- 22.4% of messages contained profanity
- 783 cursing instances
- Most popular obfuscation methods:
  1. Replacing letters
  2. Repeating letters
  3. Inserting punctuation
Our work

To identify every curse word written:

- SAPO Desporto dataset
- 10-fold cross validation
- Levenshtein edit distance
- Our extended derivation
  - 45% training set
  - 45% fitting set
  - 10% testing set
Bots
Noisy populations

- Not every user profile is filled in, or filled in correctly
- It is difficult to filter the desired population to study
- Incorrect selection of accounts may lead to incorrect or unreliable results
What is a bot?

- A “bot” is a program that posts messages on behalf of a person or organization
- Can interact with people or ignore them
- Used to post a given information at certain times

Example

Facebook estimates that as many as 60 million accounts, 2 to 3 percent of the company’s 2.07 billion regular visitors, are fakes. Sean Edgett, Twitter’s general counsel, testified before Congress that about 5 percent of its 330 million users are “false accounts or spam,” which would add up to more than 16 million fakes. (…)

Independent experts say the real numbers are far higher.

— The New York Times
The features used

- Chronological (time and period of posting)
- URLs (distribution of hosts)
- User interaction (the mentions and replies posted)
- The client application used
- The style of writing (forensic features)
The experiments

- 3 annotators
- 505 users classified, 100 messages per user
- 0.96 Fleiss’ kappa value
- 34 human, 34 bot
- 10 users/class used for training, all other users used for testing
- 50 repetitions
- SVM

![Box plot showing accuracy for different features]

- Use of URLs
- Client application
- Stylistic features
- User interaction
- Chronological features

Gustavo Laboreiro (UP)
Conclusions

- Human / bot classification can be successfully done
- Mixed accounts can murky the waters (3-way classification)
- Chronological features were not as useful as expected
- Client application was the best but poses the greatest unknowns
- Stylistic features seem a good bet
Nationality
Determining the nationality of users

- The information in the profile is not enough
- Language identification was a promising idea
- Brazilians dominated the Portuguese language microblogs
- Can we distinguish between Portuguese variants?
The features we studied

- **Stylistic features** (Quantitative, Emotion, Punctuation, Accents)
- **Named entities** (names of famous people, locations, organizations, . . . .)
- **Grammar** (gerund, “você” vs. “tu”, . . .)
- **Word tokens** (words used only on one variant)
- **URLs** (the TLD of links)

- **N-grams baseline**
The experimental setup

- Dataset based on TREC Tweets2011 corpus
- Users from “Portugal”, “Brasil” or one of the corresponding top 10 cities excluding “Porto” (can be mistaken for “Porto Alegre”)
- Exclude users with less than 100 messages
- Nationality supported by their followers (more than 10 with at least 3/4 sharing the location)
- 1400 selected randomly from each location
- Native speaker validated a 5% sample

- 5-fold cross validation
- Naïve Bayes (for speed)
Results

![Graph showing Accuracy vs Messages per user]

- **Accuracy**
  - all
  - proposed n-grams

- **Messages per user**
  - 1
  - 10
  - 100
  - 0.65
  - 0.7
  - 0.75
  - 0.8
  - 0.85
  - 0.9
  - 0.95

Gustavo Laboreiro (UP)

Noise reduction and normalization of microblogging messages

2018-05-21 40 / 44
Lessons learned

- We can distinguish between language variants
- We can get up to 95% accuracy
- Stylistic features were not a good idea
- Word tokens (lexical differences) worked the best
In closing
Final thoughts

- We contributed to a better processing of User-Generated Content
- Many problems can be expressed as classification questions
- Classification is a good tool for normalization
- Stylistic awareness can help solve several problems (but not all)
- Message “noise” may be more personal than cultural
I would like to thank...

- Supervisor and co-supervisor
- Other teaching staff
- Colleagues and friends
- SAPO Labs and REACTION group members
- Accompanying Committee
- Elements of the jury
- Department staff
- Family
Appendix
Style: Comparing the impact of the features

Feature group impact

Max F values

Feature group used

Quantitative | Emotion | Punctuation | Abbreviations | All

Gustavo Laboreiro (UP)
Adding 50 examples per problematic character reduced errors by 20%
Obfuscation: The introduction of variants

![Graph showing the relationship between variants and filtered instances]

- **Filtered**
  - False
  - True

Gustavo Laboreiro (UP)  
Noise reduction and normalization of microblog messages  
2018-05-21
Bots: experiment with 3 classes

- 3 annotators
- 545 users classified
- 0.68 Fleiss’ kappa value
- 17 cyborg accounts, 34 human, 34 bot
- 10 users/class used for training, all other users used for testing

![Accuracy chart for different features]

![Bar chart for bot, cyborg, and human categories]
Bots: The 3 classes experiment

- 4 annotators
- 174 users classified
- 0.67 Fleiss’ kappa value
- 22 users of each type selected (due to only 22 cyborg users)
- 11 users/class used for training, 11 users used for testing