What Your Username Says About You

Aaron Jaech & Mari Ostendorf
University of Washington
Motivation

- Understanding personal information in online interactions
- Why Usernames? Three reasons:
  - Expressiveness: Used to advertise oneself
    @VINCEEinNYC, @AngelTheBunny, @Gunservatively
  - Ubiquity: Twitter, Instagram, Pinterest, Snapchat, Vine, Youtube...
  - Complementary to text cues & relatively unexplored
Example

- Can you guess anything about Twitter user @mo_alq?
Example

- Can you guess anything about Twitter user @mo_alq?
- Does looking at these users help?

@moh_alsaeid       @mohamed_alattas
@mohamad_al3jmei   @mohamed_alhassn
@mohamadalarshi    @mohamed_almored
@mohamad_aljasim   @mohand_alsharif
@mohamad_alkhale   @mohd_alfozan
@mohamad_almdnee   @mohd_alsaleh1
@mohamad_almo0ha   @mohammad_al3mry
@mohamad_almsfr    
@mohamad_alrashe   
Overview

• Problem: Find out what can be inferred about an individual from only their username
  • Use gender and language identification tasks to prove the technique

• Strategy:
  • Split input into username morphemes (u-morphs)
  • Learn relationship between u-morphs and class labels
Username Morphology

- Usernames are often formed by concatenation
  \( @\text{taylorswift13} \rightarrow \text{“taylor” “swift” “13”} \)

- We use the Morfessor algorithm to build a u-morph lexicon from unsupervised data
  - Preprocess to use casing (\( \text{JohnDoe} \rightarrow \text{john$doe} \))
  - Lexicon size/morph length tuned to maximize performance on each task

- Experiments compare u-morph segmentation to character 3-gram & 4-gram (used in prior work)
Each username is just a sequence of n u-morphs
@taylorswift13 → [m₁ m₂ m₃]; m₁=taylor, m₂=swift, m₃=13
Model the relationship between u-morphs and class labels using a unigram language models

Class labels only depend on observed u-morphs
Class-dependent smoothing weights are needed when class priors are skewed

Classifier

\[
\arg\max_c \frac{p_C(c_i)}{\sum_{i=1}^n p_C(c_i)} \prod_{k=1}^n p(m_k|c_i)
\]
Gender ID

- Task: Label usernames as male/female
- Data: 44k labeled Okcupid usernames and 3.5 million unlabeled Snapchat ones, test from Okcupid
- Approach:
  - Use Snapchat data to build u-morph lexicon
  - Train male/female unigram models on Okcupid data
  - Improve models using self-training on unlabeled Snapchat data
Gender ID Results

Top Features

<table>
<thead>
<tr>
<th></th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>u-morph</td>
<td>guy, mike, matt, josh</td>
<td>girl, marie, lady, miss</td>
</tr>
<tr>
<td>trigram</td>
<td>guy, uy#, kev, joe</td>
<td>lrl, gir, grl, emm</td>
</tr>
</tbody>
</table>

Error Rates

<table>
<thead>
<tr>
<th>Features</th>
<th>Supervised</th>
<th>Self-Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-gram</td>
<td>28.7%</td>
<td>32.0%</td>
</tr>
<tr>
<td>4-gram</td>
<td>28.7%</td>
<td>29.4%</td>
</tr>
<tr>
<td>u-morph</td>
<td>27.8%</td>
<td>25.8%</td>
</tr>
</tbody>
</table>

- Top u-morph features all have a strong semantic relationship to the task.
- N-grams more prone to confusion, e.g. “guy” in “Nguyen” and “miss” in “mission”.
- Supervised learning: u-morphs have 3% reduction in error rate
- Self-training: n-grams don’t benefit but u-morph model has 10% error reduction from baseline
Language ID

• Task: Predict language of tweet from Twitter username
• Data: 540,000 usernames from 9 most popular Twitter languages.
  • Labeled by Twitter API + langid.py classifier
• Approach:
  • Build u-morph lexicon on international Twitter data
  • Train u-morph and n-gram language models
  • Average posterior probabilities from u-morph and n-gram models to create combination model
Language ID Results

4-gram model has higher recall; u-morphs give better precision. Combination model benefits from both. Multilingual u-morph lexicon less well matched to infrequent languages.

1-2% of total for each
Conclusions

- u-morphs are a good representation for username classification
- Personal characteristics can be inferred from usernames with just u-morph unigrams
  - Accuracy is good with username alone (for language ID, roughly comparable to using the whole tweet)
  - Usernames are complementary to existing features
Future Work

- Explore more tasks
- Pooled language-specific u-morph lexicons
- Improve classifier:
  - Higher-order LM
  - Other language models (e.g. log-bilinear)

Code and data available at: https://github.com/ajaech/username_analytics