CAPS: A Cross-genre Author Profiling System

Ivan Bilan and Desislava Zhekova
Center for Information and Language Processing, LMU Munich, Germany
ivan.bilan@gmx.de
zhekova@cis.uni-muenchen.de
Presentation Overview

» Overview of Author Profiling

» Training Dataset

» Software Tools

» Machine Learning Pipeline

» Custom Features

» Classification

» Final Results
Overview of Author Profiling

Author Profiling – attributing an author of a text to a certain sociodemographic class

Real world applications:

» suspect profiling in forensics
» customer-base analysis
» targeted advertising

Cross-genre author profiling:

» adaptable to any unseen genre
» label only genres that are easier to label
» merge all existing genres into one training set to overcome data scarcity
Training Dataset

PAN16 Training Set (Authors)

- English: 432
- Spanish: 249
- Dutch: 379

PAN16 Training Set (Text samples)

- English: ~200000
- Spanish: ~128000
- Dutch: ~67000

- Labelled with gender: Male Female
- Age groups: 18-24 25-34 35-49 50-64 65-xx
- Artificially increase the number of samples by labeling each text sample
- During evaluation take the most frequent prediction (or the one with the highest confidence score) for the author
Software Tools

» Python

» scikit-learn (main machine learning toolkit)

» gensim (topic modelling)

» matplotlib (visualization)

» TreeTagger (available at http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/)
  » supports part-of-speech tagging, lemmatization, stemming and chunking
  » works on multiple languages
  » has wrappers for various programming languages
  » freely available for research and education
Machine Learning Pipeline

Text Samples

Preprocessing

- Lemmatized Text
- Cleaned Text
- POS Tags

Lemma n-grams (TF-IDF)
Categorical character n-grams (TF-IDF)
Tag n-grams (TF-IDF)

Topic Modeling

Feature Selection with $\chi^2$ test

Custom Features
- (Lemmas)
- (Cleaned Text)
- (POS-based)

Training Labels

Classification Model

Test Labels
Machine Learning Pipeline

Preprocessing

» HTML and Bulletin Board Code removal

» normalization of all links to [URL]

» normalization of all usernames e.g. @username to [USER]

» duplicate sample removal

Text representations

» first experimented with stemmed text representation

» final system uses lemma and part-of-speech representation

» the results are saved in a dataframe and each feature accesses the text representation it requires
Machine Learning Pipeline

**TF-IDF - The Term Frequency-Inverse Document Frequency**

» Emphasize important words (frequent in a text, infrequent in the corpus)

Usage in CAPS:

» unigrams, bigrams, trigrams for lemmatized text
» 1-4 grams for POS text representation
» 3-grams for characters

**Topic Modelling with Latent Dirichlet Allocation (LDA)**

and Hierarchical Dirichlet Process (HDP)

» Generative statistical model that allows automated grouping of observed words into topics
» LDA requires predefined number of topics
» HDP calculates the number of topics automatically
» do not confuse with linear discriminant analysis (also known as LDA)

Usage in CAPS:

» we used LDA with 100 topics
» HDP showed decreased performance
Custom Features

» Over 40 custom features divided into the following feature clusters:

» Dictionary-based Features

» POS-Based Features

» Text Structure Features

» Stylistic Features
## Dictionary-based Features

<table>
<thead>
<tr>
<th>Feature Cluster</th>
<th>Feature Name</th>
<th>English</th>
<th>Spanish</th>
<th>Dutch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dictionary-based</td>
<td>Connective Words</td>
<td>furthermore, firstly …</td>
<td>pues, como …</td>
<td>zoals, mits …</td>
</tr>
<tr>
<td></td>
<td>Emotion Words</td>
<td>sad, bored, angry …</td>
<td>espanto, carino, calma …</td>
<td>boos, moe, zielig …</td>
</tr>
<tr>
<td></td>
<td>Contraction</td>
<td>I’d, let’s, I’ll …</td>
<td>al, del, desto …</td>
<td>m’n, ’t, zo’n …</td>
</tr>
<tr>
<td></td>
<td>Familial Words</td>
<td>wife, husband, gf …</td>
<td>esposa, esposo …</td>
<td>vriendin, man …</td>
</tr>
<tr>
<td></td>
<td>Collocations</td>
<td>dodgy, awesome, troll …</td>
<td>no manches, chido …</td>
<td>buffelen, geil …</td>
</tr>
<tr>
<td></td>
<td>Abbreviations and Acronyms</td>
<td>a.m., Inc., asap …</td>
<td>art., arch. …</td>
<td>gesch., geb. …</td>
</tr>
<tr>
<td></td>
<td>Stop Words</td>
<td>did, we, ours …</td>
<td>de, en, que …</td>
<td>van, dat, die …</td>
</tr>
</tbody>
</table>

» positive / negative sentiment lists are not used
POS-Based Features

» Use of Verbs, Interjections, Adjectives, Determiner, Conjunction, Plural Nouns

» Lexical Measure – tell how implicit or explicit the text is

\[ F = 0.5 \left( \left( \text{nouns} + \text{adjectives} + \text{prepositions} + \text{articles} \right) - \left( \text{pronouns} + \text{verbs} + \text{adverbs} + \text{interjections} \right) \right) + 100 \]

Heylighen et al. (2002)

Readability Index Formulas

» tried Automated Readability Index, SMOG Readability Formula, Flesch Reading Ease etc.

» decreased effectiveness in cross-genre setting since

» not suitable for short text samples

» e.g. Flesch Reading Ease: \( 206.835 - 1.015 \left( \frac{\text{total words}}{\text{total sentences}} \right) - 84.6 \left( \frac{\text{total syllables}}{\text{total words}} \right) \)
Text Structure Features

» Type/Token ratio
» Average word length
» Usage of punctuation marks

Stylistic features (occurrence of adjectival endings)

» English: -ly, -able, -ic, -il, -less, -ous etc.
» Spanish: -ito, -ada, -anza, -acho, -acha etc.
» Dutch: -jes, -iek, -eren etc.
Feature Scaling

**Step 1: Scale to sample length**

» the feature vector values are divided by the sample length

\[
\chi_{\text{pre-scaled}}^{(i)} = \frac{\text{feature vector value}}{\text{len(sample)}}
\]

**Step 2: Standardize**

\[
\chi_{\text{std}}^{(i)} = \frac{\chi_{\text{pre-scaled}}^{(i)} - \mu_x}{\sigma_x}
\]

» \( \chi_{\text{pre-scaled}}^{(i)} \) is a feature vector sample

» \( \mu_x \) is sample mean of the feature column

» \( \sigma_x \) represents the standard deviation of the feature column
Classification

Gender and age classified separately:

» Support Vector Machine (namely Linear Support Vector Classification) classifier used for gender classification

» Multinomial Logistic Regression for age classification
## Final Results (Cross-genre)

PAN16 Results, Accuracy (Cross-genre, all represented languages)

<table>
<thead>
<tr>
<th>PAN16</th>
<th>English</th>
<th></th>
<th></th>
<th>Spanish</th>
<th></th>
<th></th>
<th>Dutch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Gender</td>
<td>Age</td>
<td>Both</td>
<td>Gender</td>
<td>Age</td>
<td>Both</td>
<td>Gender</td>
</tr>
<tr>
<td>Best Score</td>
<td>75.64%</td>
<td>58.97%</td>
<td>39.74%</td>
<td>73.21%</td>
<td>51.79%</td>
<td>42.87%</td>
<td>61.80%</td>
</tr>
<tr>
<td>CAPS</td>
<td>74.36%</td>
<td>44.87%</td>
<td>33.33%</td>
<td>62.50%</td>
<td>46.43%</td>
<td>37.50%</td>
<td>55.00%</td>
</tr>
<tr>
<td>Lowest Score</td>
<td>46.15%</td>
<td>32.05%</td>
<td>14.10%</td>
<td>46.43%</td>
<td>21.43%</td>
<td>21.43%</td>
<td>41.60%</td>
</tr>
</tbody>
</table>

### Final Top 5 Ranking (PAN16, by overall average)

<table>
<thead>
<tr>
<th>Place</th>
<th>1st</th>
<th>2nd</th>
<th>3rd (CAPS)</th>
<th>4th</th>
<th>5th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result</td>
<td>52.58%</td>
<td>52.47%</td>
<td>48.34%</td>
<td>46.02%</td>
<td>45.93%</td>
</tr>
</tbody>
</table>
Final Results (Single genre)

» the system also performs rather effectively in single genre setting

PAN14 and PAN15 Results, Accuracy (Single genre, English)

<table>
<thead>
<tr>
<th>PAN14-15</th>
<th>Twitter (PAN15)</th>
<th>Blogs (PAN14)</th>
<th>Hotel Reviews (PAN14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Gender</td>
<td>Age</td>
<td>Gender</td>
</tr>
<tr>
<td>Best Score</td>
<td>85.92%</td>
<td>83.80%</td>
<td>67.95%</td>
</tr>
<tr>
<td>CAPS</td>
<td>81.69%</td>
<td>73.24%</td>
<td>66.67%</td>
</tr>
</tbody>
</table>
Future work

» use dependancy parsing and extract features based on the tree representation

» improve features for Spanish and Dutch
Thank you for your attention!
References