Mining E-mail Content for Author Identification Forensics

O. de Vel, A. Anderson, M. Corney and G. Mohay

A presentation by Fabian Duffhauß

Reasons for Author Identification of E-mails

- Everyday 200 billions of e-mails are sent
  → 90 % spam
- Misuse of e-mails:
  - Distribute inappropriate messages or documents
  - Send offensive or threatening material
- sender try to hide their identity

→ identify the author of e-mail misuse

E-mail Topic and Authors Used in the Experiments

Topic Category	*Author Category AC_i (i = 1; 2; 3)*			*Topic Total*
	Author AC_1	Author AC_2	Author AC_3	
Movie	15	21	21	59
Food	12	21	25	58
Travel	3	21	15	39
Author Total	30	63	63	156

- salutations, reply text, attachments and signatures are removed
- Existence and position are stored

170 Style Marker Attribute Types

- Number of blank lines/total number of lines
- Average sentence length
- Average word length (number of characters)
- Vocabulary richness i.e., V/M
- Total number of function words/M
- Function word frequency distribution (122 features)
- Total number of short words/M
- Count of hapax legomena/M
- Count of hapax legomena/V
- Total number of characters in words/C
- Total number of alphabetic characters in words/C
- Total number of upper-case characters in words/C
- Total number of digit characters in words/C
- Total number of white-space characters/C
- Total number of space characters/C
- Total number of space characters/number white-space characters
- Total number of tab spaces/C
- Total number of tab spaces/number white-space characters
- Total number of punctuations/C
- Word length frequency distribution/M (30 features)

M = total number of words

V = total number of distinct words

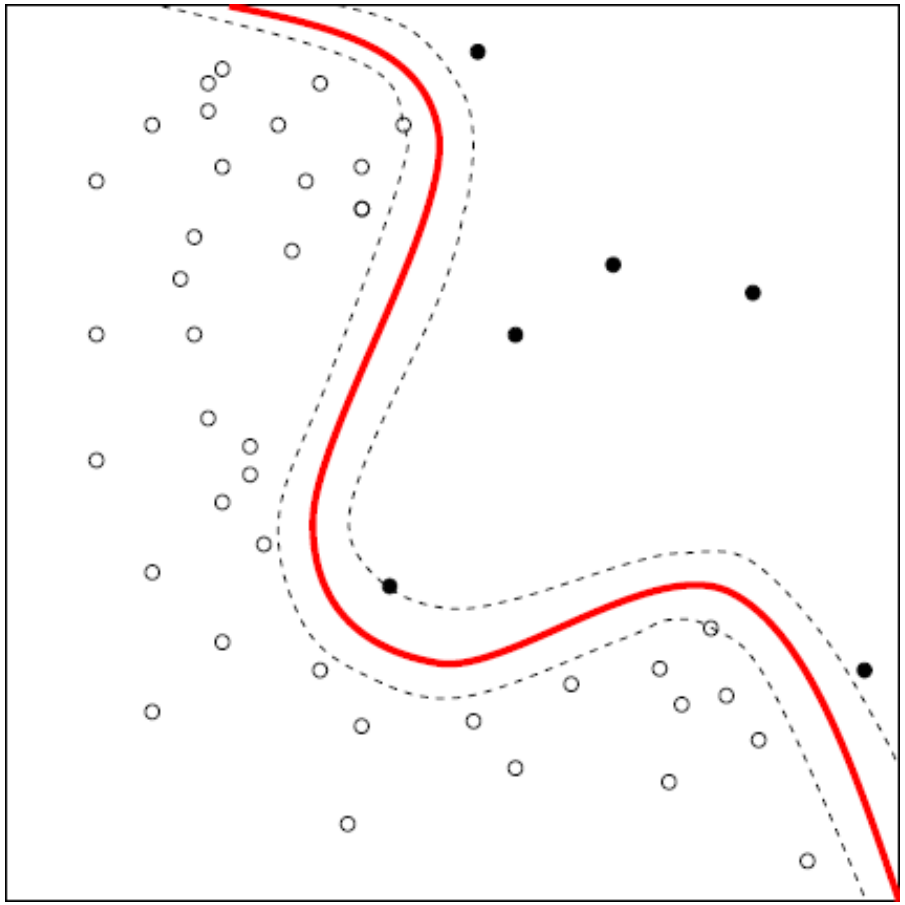
C = total number of characters

21 Structural Attribute Types

- Has a greeting acknowledgment
- Uses a farewell acknowledgment
- Contains signature text
- Number of attachments
- Position of requoted text within e-mail body
- HTML tag frequency distribution/total number of HTML tags (16 features)

Support Vector Machine Classifier

- SVM^{light}
- separate objects into two different classes.
- Best results with a polynomial kernel of degree 3



Measuring Units

- *C* = set of objects that belong to a class
- *A* = set of objects the classifier has identified as belonging to the class

$$recall\ R = \frac{\|C \cap A\|}{\|C\|} \qquad precision\ P = \frac{\|C \cap A\|}{\|A\|}$$

$$F = \frac{2RP}{R + P}$$

First Experiment

- Mixed topics
- Stratified 10-fold cross validation procedure

style markers and structural features

Performance Statistic	*Author Category, AC_i (i = 1, 2, 3)*		
	Author AC_1	*Author AC_2*	*Author AC_3*
P_{AC_i}	100.0 %	83.8 %	93.8 %
R_{AC_i}	63.3 %	98.3 %	89.6 %
F_{AC_i}	77.6 %	90.5 %	91.6 %

only style markers

Performance Statistic	*Author Category, AC_i (i = 1, 2, 3)*		
	Author AC_1	*Author AC_2*	*Author AC_3*
P_{AC_i}	100.0 %	93.0 %	83.6 %
R_{AC_i}	60.0 %	80.3 %	93.3 %
F_{AC_i}	75.0 %	86.2 %	88.2 %

Second Experiment

- Training set: E-mails with topic “Movie”

style markers and structural features

Topic Class	Author Category, AC_i (i = 1, 2, 3)								
	Author AC_1			Author AC_2			Author AC_3		
	P_{AC_1}	R_{AC_1}	F_{AC_1}	P_{AC_2}	R_{AC_2}	F_{AC_2}	P_{AC_3}	R_{AC_3}	F_{AC_3}
Food	100.0	16.7	28.6	77.8	100.0	87.5	85.2	92.0	88.5
Travel	100.0	33.3	50.0	90.9	100.0	95.2	100.0	100.0	100.0

categorisation performance results (in %)

Third Experiment

- Number of function words: 320 (instead of 122)
  - Split into parts-of-speech words and others

- Result: No improvements

PAN-11 Author Identification Training Corpus

training sets

Name	Number of Authors	Number of Documents
Large	72	9337
Small	26	3001
Verify1	1	42
Verify2	1	55
Verify3	1	47

Validation sets

Name	Number of Authors	Number of Documents
LargeValid	66	1298
LargeValid+	86	1440
SmallValid	23	518
SmallValid+	43	601
Verify1Valid+	24	104
Verify2Valid+	21	95
Verify3Valid+	23	100

Live Demonstration

- Parser in C++:
  - Reads a list of function words
  - Reads the e-mail bodies
  - Extracts style marker attributes
  - Creates training and test files
- SVM^{light}-Learn:
  - Reads the training file
  - Creates a model
- SVM^{light}-Classify:
  - Reads the model and the test file
  - Makes a prediction