Author Obfuscation

Attacking the State of the Art in Authorship Verification

Martin Potthast, Matthias Hagen, and Benno Stein
Bauhaus-Universität Weimar

www.webis.de
Alice

is known
to have
written
Obfuscation vs. Identification

Introduction

Alice is known to have written
Obfuscation vs. Identification

Introduction

Alice

is known to have written

Eve
Alice is known to have written

is used as reference

Eve automatically verifies authorship

is subject to analysis
Obfuscation vs. Identification

Introduction

Alice is known to have written automatically obfuscates text

Eve automatically verifies authorship is used as reference
Obfuscation vs. Identification

Introduction

Alice is known to have written automatically obfuscates text

Eve automatically verifies authorship

is used as reference is subject to analysis
Obfuscation vs. Identification

Introduction

Alice is known to have written automatically obfuscates text

Eve automatically verifies authorship

is used as reference

circumvents OR obstructs

is subject to analysis

© Potthast 2016
Obfuscation vs. Identification

Introduction

Alice is known to have written automatically obfuscates text

Eve automatically verifies authorship is used as reference is subject to analysis

Alice is known to have written

is used as reference

Eve automatically verifies authorship

is subject to analysis

Author masking:
Given two documents by the same author, paraphrase the designated one so that the author cannot be verified anymore.

Authorship verification:
Given two documents, decide whether they have been written by the same author.
Obfuscation vs. Identification

Introduction

Alice

is known to have written

automatically obfuscates text

Eve

automatically verifies authorship

is subject to analysis

circumvents or obstructs

is used as reference

Key questions

- How vulnerable are verification approaches to obfuscation?
- How successful are obfuscation approaches against verification?
- Which technology will ultimately dominate the other?
We call an obfuscation software

- **safe**, if its obfuscated texts can not be attributed to their original authors,
- **sound**, if its obfuscated texts are textually entailed by their originals, and
- **sensible**, if its obfuscated texts are well-formed and inconspicuous.
Obfuscation Evaluation

Taxonomy of Evaluation Dimensions

- **Safety**
  - Author identification methods
    - Manual
    - Automatic
  - De-obfuscation attacks
    - Manual

- **Soundness**
  - Manual safety evaluation against forensic linguists not scalable
  - Automatic safety evaluation requires large amount of implementations
  - Several obfuscation approaches can be undone

- **Sensibleness**
Obfuscation Evaluation

Taxonomy of Evaluation Dimensions

- Safety
  - Author identification methods
    - Manual
    - Automatic
  - De-obfuscation attacks
    - Manual

- Soundness
  - Paraphrasis (strict)
    - Manual
    - Automatic
  - Textual entailment (relaxed)
    - Manual
    - Automatic

- Sensibleness

- Paraphrase: obfuscation restates the original with different words
- Textual entailment: obfuscation follows logically from original
- Support manual review with visual text comparison
Obfuscation Evaluation

Taxonomy of Evaluation Dimensions

- Safety
  - Author identification methods
    - Manual
    - Automatic
  - De-obfuscation attacks
    - Manual

- Soundness
  - Paraphrasis (strict)
    - Manual
    - Automatic
  - Textual entailment (relaxed)
    - Manual
    - Automatic

- Sensibleness
  - Grammaticality (strict); Readability (relaxed)
    - Manual
    - Automatic
  - Obfuscation detection
    - Manual
    - Automatic

- Obfuscation Evaluation

- Safety
- Soundness
- Sensibleness

- Relax grammaticality: machine translation also not perfect, yet useful
- Hiding obfuscation useful to avoid in-depth (manual) forensic analysis
- Automatic evaluation involves cutting edge research

© Potthast 2016
Obfuscation Evaluation
Taxonomy of Evaluation Dimensions

- **Safety**
  - Author identification methods
    - Manual
    - Automatic
  - De-obfuscation attacks
    - Manual

- **Soundness**
  - Paraphrasis (strict)
    - Manual
    - Automatic
  - Textual entailment (relaxed)
    - Manual
    - Automatic

- **Sensibleness**
  - Grammaticality (strict); Readability (relaxed)
    - Manual
    - Automatic
  - Obfuscation detection
    - Manual
    - Automatic

- Evaluations conducted in our shared task
Obfuscation Evaluation

Shared Task Setup
Obfuscation Evaluation

Shared Task Setup

PAN 13/14/15: Authorship Verification Evaluation

TP  FN
Obfuscation Evaluation

Shared Task Setup

PAN 13/14/15: Authorship Verification Evaluation

- TP
- FN
- TN
- FP
Obfuscation Evaluation

Shared Task Setup

<table>
<thead>
<tr>
<th>PAN 13/14/15: Authorship Verification</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAN 16: Author Masking</td>
<td>Evaluation (safety)</td>
</tr>
</tbody>
</table>

This setup tells us

- whether an obfuscator can defeat a verifier
Obfuscation Evaluation

Shared Task Setup

This setup tells us

- whether an obfuscator can defeat a verifier
- whether an obfuscator can defeat a verifier in general
Obfuscation Evaluation

Shared Task Setup

This setup tells us

- whether an obfuscator can defeat a verifier
- whether an obfuscator can defeat a verifier in general
- whether an obfuscator can defeat verifiers in general
Obfuscation Evaluation

Shared Task Setup

<table>
<thead>
<tr>
<th>PAN 13/14/15: Authorship Verification</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAN 16: Author Masking</td>
<td>Evaluation (safety)</td>
</tr>
</tbody>
</table>

This setup tells us

- whether an obfuscator can defeat a verifier
- whether an obfuscator can defeat a verifier in general
- whether an obfuscator can defeat verifiers in general
- whether obfuscators can defeat verifiers in general
Obfuscation Evaluation
Measuring Obfuscation Impact

Performance without obfuscation

\[
\begin{array}{cc}
TN_1 & FP_1 \\
TP_1 & FN_1 \\
\end{array}
\]

Performance with obfuscation

\[
\begin{array}{cc}
TN_2 & FP_2 \\
TP_2 & FN_2 \\
\end{array}
\]
Obfuscation Evaluation
Measuring Obfuscation Impact

Perfomance without obfuscation

\[
\begin{array}{cc}
TN_1 & FP_1 \\
TP_1 & FN_1 \\
\end{array}
\]

Performance with obfuscation

\[
\begin{array}{cc}
TN_2 & FP_2 \\
TP_2 & FN_2 \\
\end{array}
\]
### Obfuscation Evaluation

**Measuring Obfuscation Impact**

<table>
<thead>
<tr>
<th></th>
<th>TP1</th>
<th>FN1</th>
<th>TN1</th>
<th>FP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>without obfuscation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>TP2</th>
<th>FN2</th>
<th>TN2</th>
<th>FP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>with obfuscation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© Potthast 2016
Obfuscation Evaluation

Measuring Obfuscation Impact

Performance without obfuscation

Performance with obfuscation

\[
\begin{array}{cc}
\text{TN}_1 & \text{FP}_1 \\
\text{TP}_1 & \text{FN}_1 \\
\end{array}
\]

\[
\begin{array}{cc}
\text{TN}_2 & \text{FP}_2 \\
\text{TP}_2 & \text{FN}_2 \\
\end{array}
\]
Side effects indicate that the verifier employs corpus-relative features.
Corpus-relative features are an anti-pattern since verification cases do not come in groups.
Obfuscation Evaluation
Measuring Obfuscation Impact

\[ \text{rec}_1 = \frac{\text{TP}_1}{\text{TP}_1 + \text{FN}_1} \]

\[ \text{rec}_2 = \frac{\text{TP}_2}{\text{TP}_2 + \text{FN}_2} \]

\[ \Delta_{\text{rec}} = \text{rec}_2 - \text{rec}_1 \]
Obfuscation Evaluation
Measuring Obfuscation Impact

Performance without obfuscation

\[
\begin{array}{cc}
\text{TN}_1 & \text{FP}_1 \\
\text{TP}_1 & \text{FN}_1 \\
\end{array}
\]

Performance with obfuscation

\[
\begin{array}{cc}
\text{TN}_2 & \text{FP}_2 \\
\text{TP}_2 & \text{FN}_2 \\
\end{array}
\]

impact

side effects

\[
\text{imp} = \begin{cases} 
- \frac{\Delta_{	ext{rec}}}{\text{rec}_1} & \text{if } \Delta_{	ext{rec}} < 0, \\
- \frac{\Delta_{	ext{rec}}}{1 - \text{rec}_1} & \text{else.}
\end{cases}
\]
## Safety Evaluation Results

<table>
<thead>
<tr>
<th>Obfuscator</th>
<th>Dataset</th>
<th>Pos. cases</th>
<th>avg $\Delta_{rec}$</th>
<th>avg imp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mihaylova et al.</td>
<td>PAN13</td>
<td>14</td>
<td>-0.2778</td>
<td>0.4690</td>
</tr>
<tr>
<td>Keswani et al.</td>
<td>PAN13</td>
<td>14</td>
<td>-0.2361</td>
<td>0.4245</td>
</tr>
<tr>
<td>Mansoorizadeh et al.</td>
<td>PAN13</td>
<td>14</td>
<td>-0.0933</td>
<td>0.1442</td>
</tr>
<tr>
<td>Mihaylova et al.</td>
<td>PAN14 EE</td>
<td>100</td>
<td>-0.2304</td>
<td>0.4891</td>
</tr>
<tr>
<td>Keswani et al.</td>
<td>PAN14 EE</td>
<td>100</td>
<td>-0.1873</td>
<td>0.4058</td>
</tr>
<tr>
<td>Mansoorizadeh et al.</td>
<td>PAN14 EE</td>
<td>100</td>
<td>-0.1038</td>
<td>0.2512</td>
</tr>
<tr>
<td>Mihaylova et al.</td>
<td>PAN14 EN</td>
<td>100</td>
<td>-0.2456</td>
<td>0.4750</td>
</tr>
<tr>
<td>Keswani et al.</td>
<td>PAN14 EN</td>
<td>100</td>
<td>-0.1783</td>
<td>0.3769</td>
</tr>
<tr>
<td>Mansoorizadeh et al.</td>
<td>PAN14 EN</td>
<td>100</td>
<td>-0.0958</td>
<td>0.2345</td>
</tr>
<tr>
<td>Mihaylova et al.</td>
<td>PAN15</td>
<td>250</td>
<td>-0.2009</td>
<td>0.3649</td>
</tr>
<tr>
<td>Keswani et al.</td>
<td>PAN15</td>
<td>250</td>
<td>-0.1298</td>
<td>0.2543</td>
</tr>
<tr>
<td>Mansoorizadeh et al.</td>
<td>PAN15</td>
<td>250</td>
<td>-0.0994</td>
<td>0.1952</td>
</tr>
</tbody>
</table>
Obfuscation Approaches
Targets 6 style features: sentence length, punctuation, stop words, parts of speech, all caps, word frequencies

- Computation of expected values based on background corpus
- Obfuscation towards the average using rule-based text operations; 14 rules

Observations

- Unfitting replacements, semantic distortions, overdone error insertion
Obfuscation Approaches
Keswani et al.

- Round-trip translation English – German – French – English
- Based on Moses SMT toolkit, trained on Europarl corpus

Observations
- Fragments of non-English text remain from translation
- Europarl corpus ill-suited for the genres of the test datasets
- Text unsound and unreadable
Obfuscation Approaches
Mansoorizadeh et al.

run-time system organization called the JRE. This approach has some advantages and disadvantages and it is worth comparing these three options in order to appreciate the implications for the Java coffee developer. The compiler translates the source code into machine code for the relevant hardware/OS combination. Strictly speaking there are two stages: compilation of program units (usually files), followed by ‘linking’ when the complete accomplishment

- Conservative paraphrasing: at most 1 word per sentence replaced
- Replacement candidates chosen among 200 most frequent words
- Replacements obtained from WordNet, based on word similarity
- Selection based on commonness under 4-gram language model

Observations
- Lots of problems with genre-specific terminology
Obfuscation Evaluation

Peer Evaluation

- Given training runs of each obfuscator, evaluate their performance
- Participants anonymized; yet, self-identification likely
- 3 participating teams, 1 independent (i.e., without obfuscator submission): Mihaylova et al., Mansoorizadeh et al., and Liebeck et al.
Obfuscation Evaluation

Peer Evaluation

- Given training runs of each obfuscator, evaluate their performance
- Participants anonymized; yet, self-identification likely
- 3 participating teams, 1 independent (i.e., without obfuscator submission): Mihaylova et al., Mansoorizadeh et al., and Liebeck et al.

Safety

- In all cases, based on GLAD verification system
- 2 rank Mihaylova $\succ$ Keswani $\succ$ Mansoorizadeh (in agreement with us)
- Mansoorizadeh ranks Keswani $\succ$ Mansoorizadeh $\succ$ Mihaylova
Obfuscation Evaluation

Peer Evaluation

- Given training runs of each obfuscator, evaluate their performance
- Participants anonymized; yet, self-identification likely
- 3 participating teams, 1 independent (i.e., without obfuscator submission): Mihaylova et al., Mansoorizadeh et al., and Liebeck et al.

Safety

- In all cases, based on GLAD verification system
- 2 rank Mihaylova ≻ Keswani ≻ Mansoorizadeh (in agreement with us)
- Mansoorizadeh ranks Keswani ≻ Mansoorizadeh ≻ Mihaylova

Soundness

- Liebeck employ SemEval STS measure, scoring from 5 (best) to 0 (worst)
- All obfuscators score >4
- Mihaylova and Mansoorizadeh assign scores manually
- 2 rank Mansoorizadeh ≻ Mihaylova ≻ Keswani (in agreement with us)
- Mihaylova ranks Mihaylova ≻ Mansoorizadeh ≻ Keswani
Obfuscation Evaluation
Peer Evaluation (cont’d)

Sensibleness

- All assessors resort to manual review
- Scales include 0-5 (Mansoorizadeh), 0-2 (Liebeck), 0,1 (Mihaylova)
- All rank Mansoorizadeh $\succ$ Mihaylova $\succ$ Keswani (in agreement with us)
Obfuscation Evaluation

Peer Evaluation (cont’d)

Sensibleness

- All assessors resort to manual review
- Scales include 0-5 (Mansoorizadeh), 0-2 (Liebeck), 0,1 (Mihaylova)
- All rank Mansoorizadeh ≻ Mihaylova ≻ Keswani (in agreement with us)

Organizer Evaluation wrt. Soundness and Sensibleness

Sensibleness

- School grading 1-5 (fail) of 1 text per obfuscator per dataset
- Our ranking: Mansoorizadeh (2) ≻ Mihaylova (4-5) ≻ Keswani (5)

Soundness

- After grading, reading the originals, and comparison with Diff visualization
- Judgment of whole texts on scale 0 (“incorrect”), 1 (“passable”), 2 (“correct”)
- Our ranking: Mansoorizadeh (1+) ≻ Mihaylova (0+) ≻ Keswani (0)
Conclusion

Summary

- 3 author obfuscators vs. 44 authorship verifiers in 4 settings
- Authorship verifiers represent the state of the art as per PAN’13/14/15
- Obfuscators flip on average from 20% up to 49% of true positive decisions
- Even conservative obfuscation has significant impact
Conclusion

Summary

- 3 author obfuscators vs. 44 authorship verifiers in 4 settings
- Authorship verifiers represent the state of the art as per PAN’13/14/15
- Obfuscators flip on average from 20% up to 49% of true positive decisions
- Even conservative obfuscation has significant impact

Take-away messages

- State of the art in authorship verification vulnerable to obfuscation
- Automatic obfuscation is feasible, yet far from perfection
- Hardly any authorship researcher considers obfuscation a threat
- Better ways of assessing soundness and sensibleness at scale needed
Conclusion

Summary

- 3 author obfuscators vs. 44 authorship verifiers in 4 settings
- Authorship verifiers represent the state of the art as per PAN’13/14/15
- Obfuscators flip on average from 20% up to 49% of true positive decisions
- Even conservative obfuscation has significant impact

Take-away messages

- State of the art in authorship verification vulnerable to obfuscation
- Automatic obfuscation is feasible, yet far from perfection
- Hardly any authorship researcher considers obfuscation a threat
- Better ways of assessing soundness and sensibleness at scale needed
- Author obfuscation and author identification are locked in an instance of the “Potter-Voldemort Conundrum:”

*Neither can live while the other survives*
Conclusion

Summary

- 3 author obfuscators vs. 44 authorship verifiers in 4 settings
- Authorship verifiers represent the state of the art as per PAN’13/14/15
- Obfuscators flip on average from 20% up to 49% of true positive decisions
- Even conservative obfuscation has significant impact

Take-away messages

- State of the art in authorship verification vulnerable to obfuscation
- Automatic obfuscation is feasible, yet far from perfection
- Hardly any authorship researcher considers obfuscation a threat
- Better ways of assessing soundness and sensibleness at scale needed

- Author obfuscation and author identification are locked in an instance of the “Potter-Voldemort Conundrum:”

  *Neither can live while the other survives*

Thank you for your attention!