Who wrote the web?

KOPPEL, WINTER (2014): DETERMINING IF TWO DOCUMENTS ARE WRITTEN BY THE SAME AUTHOR

Tolga Buz

Introduction

- Many online documents are written pseudonymously or anonymously
- Authorship often of financial or legal importance
  - e.g. several product reviews by same author?
  - or two threatening letters?
  - or many students‘ homework?

=> How can we solve the verification problem?

Authorship verification problem

- open-set problem:

  Is an anonymous document written by a given candidate author or someone else?

- Usually we have writing samples from each author

- “If we can determine if any two documents are written by the same author, we can solve any […] standard authorship attribution problem.”

=> We compare the anonymously written document with writing samples of each candidate.

Solution outline

- Documents X and Y are to be compared
- Produce a set of “impostor“ documents
- Ask if X is “sufficiently more similar to Y than to any of the generated impostors“
- Use proper methods to select impostors
- Measurement of similarity: randomly selecting subsets of features
- Works suprisingly good, even on documents with 500 words

Experimental Setup

- Based on several thousand bloggers‘ output
- Pairs of (fragments of) blog posts <X, Y>
- X: blogger‘s first 500 words
- Y: (different) blogger‘s last 500 words

→ 500 words = relatively **short** document

- Corpus = 500 pairs
- half corpus: both from same blogger
- other half: each from a different blogger

Similarity-Based Baseline Method

- Measure similarity & if above threshold: assing to class *same-author*
  - like *Abbasi & Chen (2008): "similarity detection"*, but with simpler features
- **document** = numerical vector (100,000 values)
  - Each value = frequency of space-free "character 4-gram"
- space free **character 4-gram** = string of 4 characters
  (or fewer chars, surrounded by spaces)
- 100,000 most frequent features stored in vector

Similarity-Based Baseline Method

- 4-grams are much simpler than other feature sets
- Still at least as effective
- Main advantage: **very large & homogenous feature set**
- Two similarity measures used:

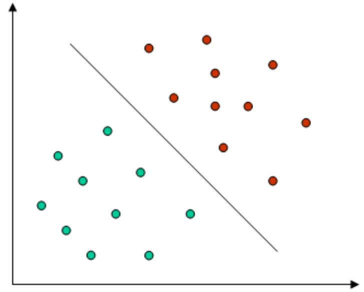
$$\mathrm{sim}(X,Y) = \mathrm{cosine}(\vec{X},\vec{Y}) = \vec{X} * \vec{Y} / \|\vec{X}\| * \|\vec{Y}\|$$

$$\mathrm{sim}(X,Y) = \mathrm{minmax}(\vec{X},\vec{Y}) = \frac{\sum_{i=1}^{n} \min(x_i, y_i)}{\sum_{i=1}^{n} \max(x_i, y_i)}$$

- best accuracy: 70.6% (cosine) resp. 74.2% (minmax)
- Disadvantage: ignores factors like genre, topic, etc.

Supervised Baseline Method

- training set: 1,000 pairs <X, Y>
- labeled as *different-author pair* or *same-author pair*
- **Supervised methods**
  - 1. Calculate: $\mathrm{diff}(X, Y) = \langle |x_1 - y_1|, \ldots, |x_n - y_n| \rangle$
  - 2. label $\mathrm{diff}(X, Y)$ same as <X, Y> ⟶ *different-author* or *same-author*
  - 3. Use labeled examples as training examples ⟶ Support Vector Machine (SVM)

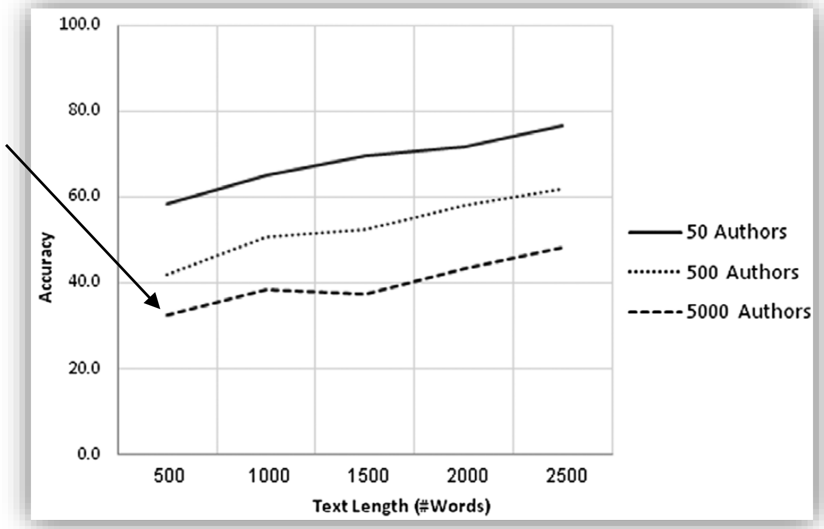


- Accuracy = 79.6%

Source: https://de.wikipedia.org/wiki/Support_Vector_Machine

Many-Candidates Problem

- Many candidate authors for an anonymous document
  ⟶ open-set identification problem
- Setup: 5,000 bloggers‘ first 500 words & last 500 words from anonymous
  ⟶ snippet
- Measure similarity with min-max
- Accuracy for 5,000 authors and 500 words: **32.5%**



⟶ not enough for most applications

Many-Candidates Problem

- What if the author is not in the set?
- Only using a similarity threshold is not enough
- We need to **vary the feature sets:**

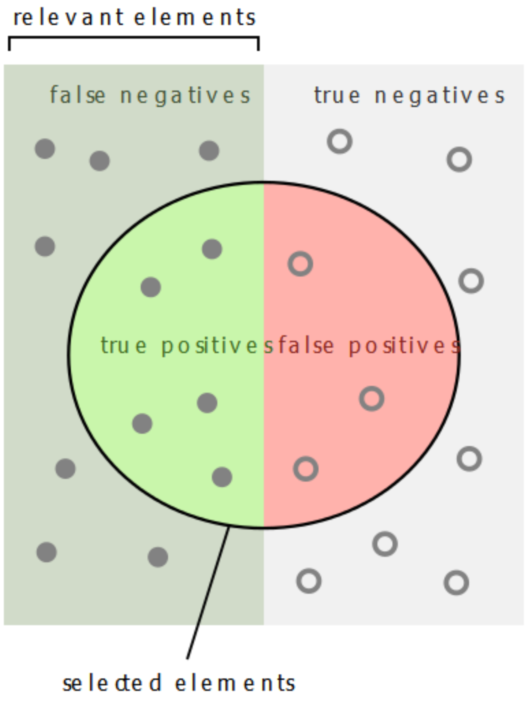
Given: a snippet to be assigned; known-texts for each of C candidates

1. **Repeat** k times
   a. Randomly choose half of the features in the full feature set.
   b. Find top known-text match to snippet using min-max similarity
2. **For each** candidate author A,
   a. Score(A) = proportion of times A is top match

Output: argmax_A Score(A) **if** max Score(A) > σ*; else Don't Know

Many-Candidates Results

- **k=100 iterations** is sufficient
- **Threshold σ*** can be varied to obtain **recall-precision tradeoff** (here: σ* = 0.80)



Source:
https://en.wikipedia.org/wiki/Precision_and_recall

- For 500 candidates:
  - 90.2% precision &
  - 22.2% recall
  - From 1,000 snippets that belong to none:
    94.5% correctly (not-)attributed

The Impostors Method

- Many-Candidates Problem can be solved well
- Impostors help reduce the verification problem to many-candidates

1. Generate a set of impostors $Y_1, \ldots, Y_m$ (as specified below).
2. Compute $score_X(Y)$ = the number of choices of feature sets (out of 100) for which $sim(X, Y) > sim(X, Y_i)$, for all $i = 1, \ldots, m$.
3. Repeat the above with impostors $X_1, \ldots, X_m$ and compute $score_Y(X)$ in an analogous manner.
4. If average($score_X(Y)$, $score_Y(X)$ is greater than a threshold σ^*, assign $\langle X, Y \rangle$ to *same-author*.

The Impostors Method

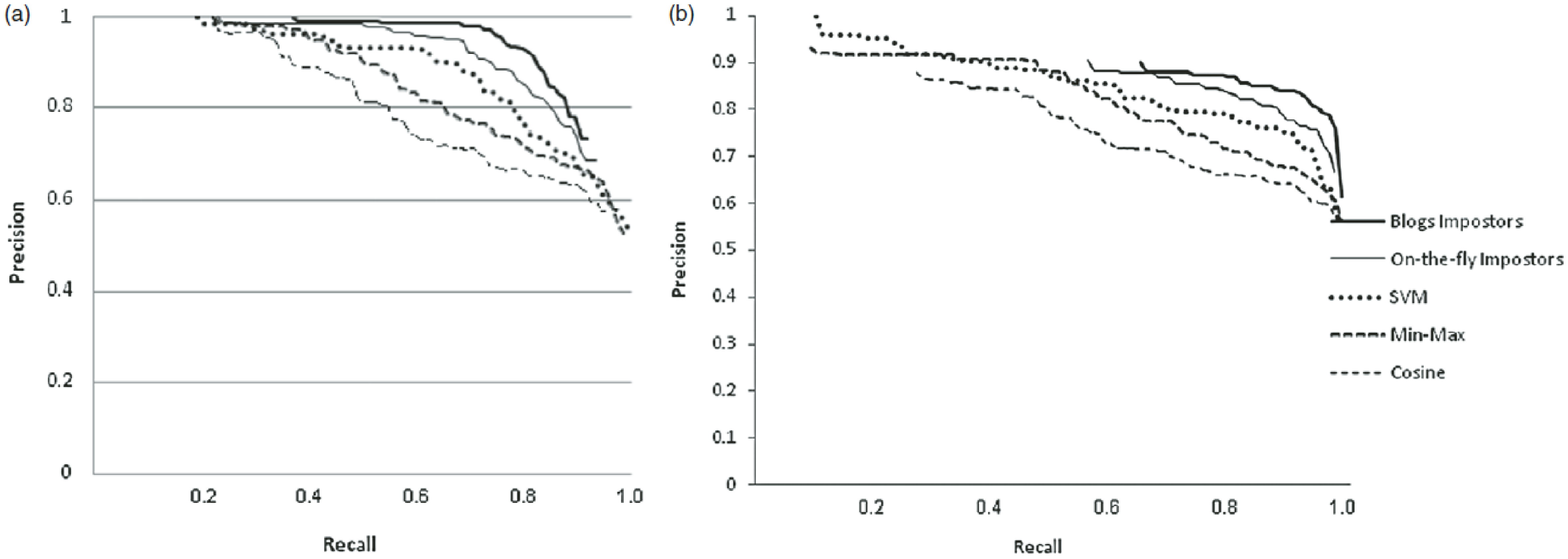
- Correct choice of impostors is critical
- Wrong choice can give too many false negatives or false positives

=> We need an optimal combination of:

Impostor quality, quantity and score threshold.

- Three methods of generating potential impostors for Y:
  - **Fixed:** fixed set, no special relation to the document
  - **On-the-fly:** variety of small random sets, use in Google query & aggregate top results
  - <u>**Blogs:** choose texts from other bloggers in same genre</u> => best recall & precision

Results


(a)
Precision
1
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0
Recall
(b)
Precision
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
1.0
Recall
Blogs Impostors
On-the-fly Impostors
SVM
Min-Max
Cosine

Ranking

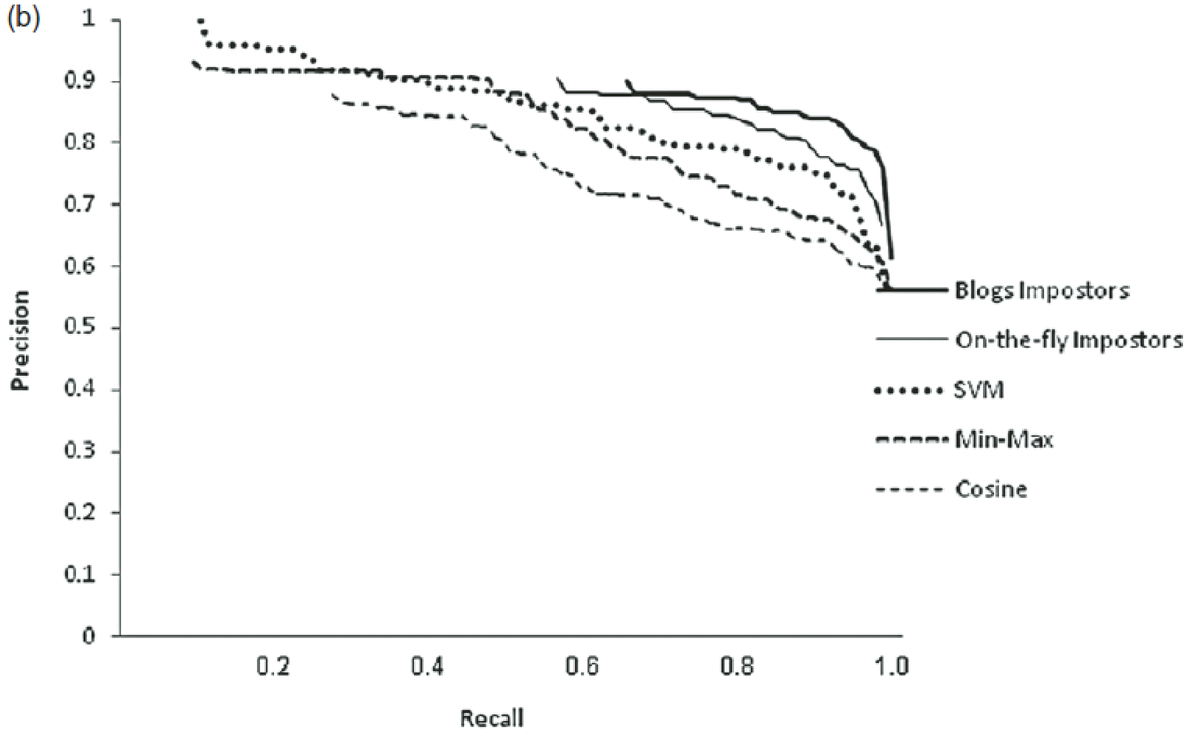
1. Impostors method using “blog universe“
(accuracy: 87.4%)

2. Impostors method using “On-the-fly universe“
(83.2%)

3. SVM classifier learned from training set
(approx. 80%)

4. Similarity using min-max
(approx. 75%)

5. Similarity using cosine
(approx. 70%)



Conclusions – Pro & Con

+ Almost unsupervised impostors method works pretty good

+ Able to give good results with very short texts (≥500 words)

+ Can be applied to many real-life problems

- Bad choice of impostors can heavily influence the results

- Impostors must not contain any text from “our“ authors

- Hard to rely on, if topic and genre differ