Technology for Text Plagiarism Analysis

Benno Stein + the webis Group
Bauhaus-Universität Weimar
www.webis.de

Outline

- Overview

- Plagiarim Corpus
- Detection Performance Measures

- Heuristic Retrieval
- Hash-based Search

- Intrinsic Detection and Authorship Verification
- Post-Processing with Unmasking

- Cross Language Analysis
- Knowledge-based Post Processing

- Competition on Plagiarim Detection
- Software

Overview

Overview

Plagiarism is the practice of claiming, or implying, original authorship of someone else's written or creative work, in whole or in part, into one's own without adequate acknowledgment.

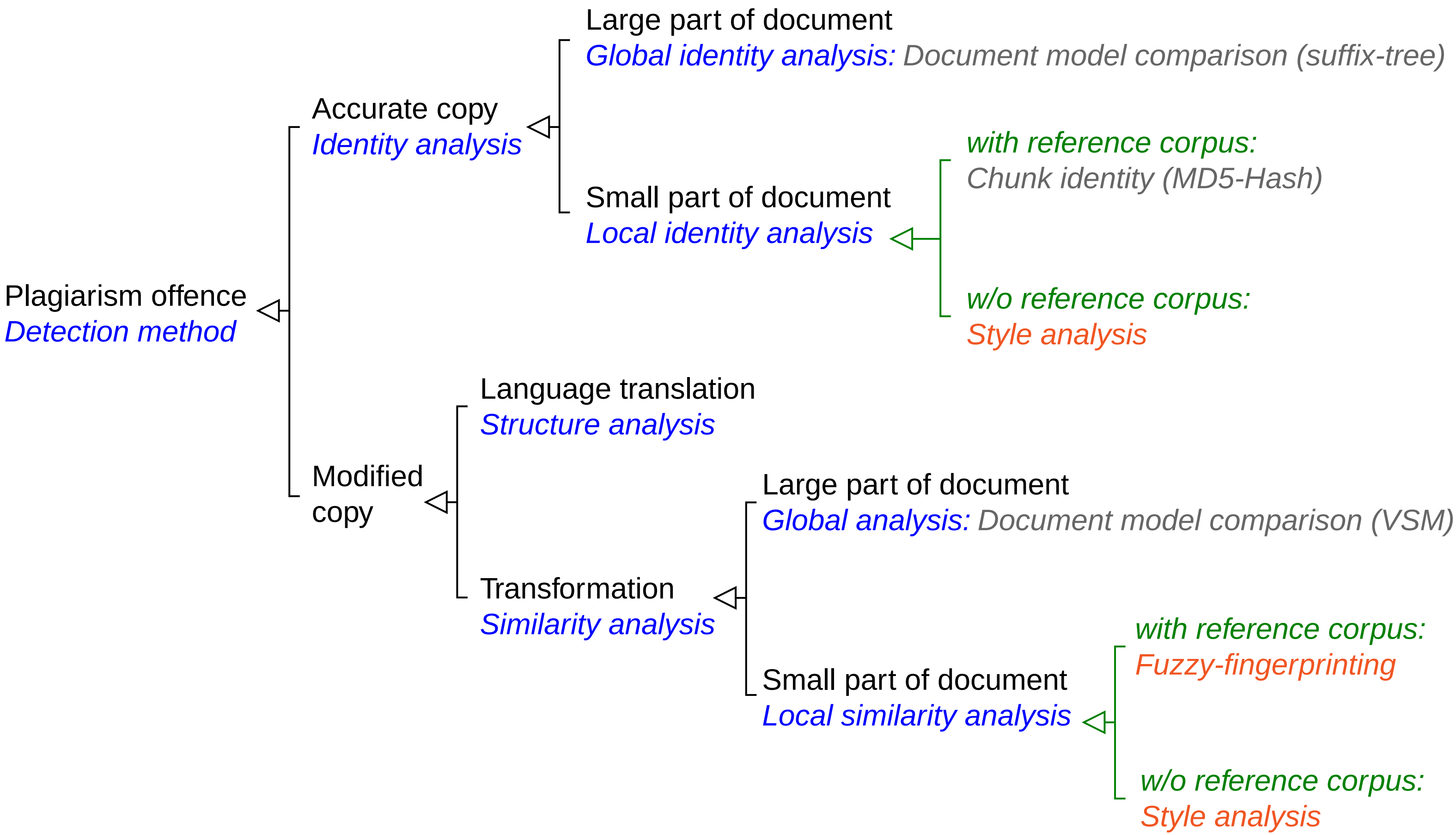
[Wikipedia: Plagiarism]

- Plagiarism is observed in literature, music, software, scientific articles, newspaper, advertisement, Web sites, etc.
- A study among 18 000 university students in the United States shows that almost 40% of them have plagiarized at least once. [1]

[1] D. McCabe. Research Report of the Center for Academic Integrity. `http://www.academicintegrity.org`, 2005.

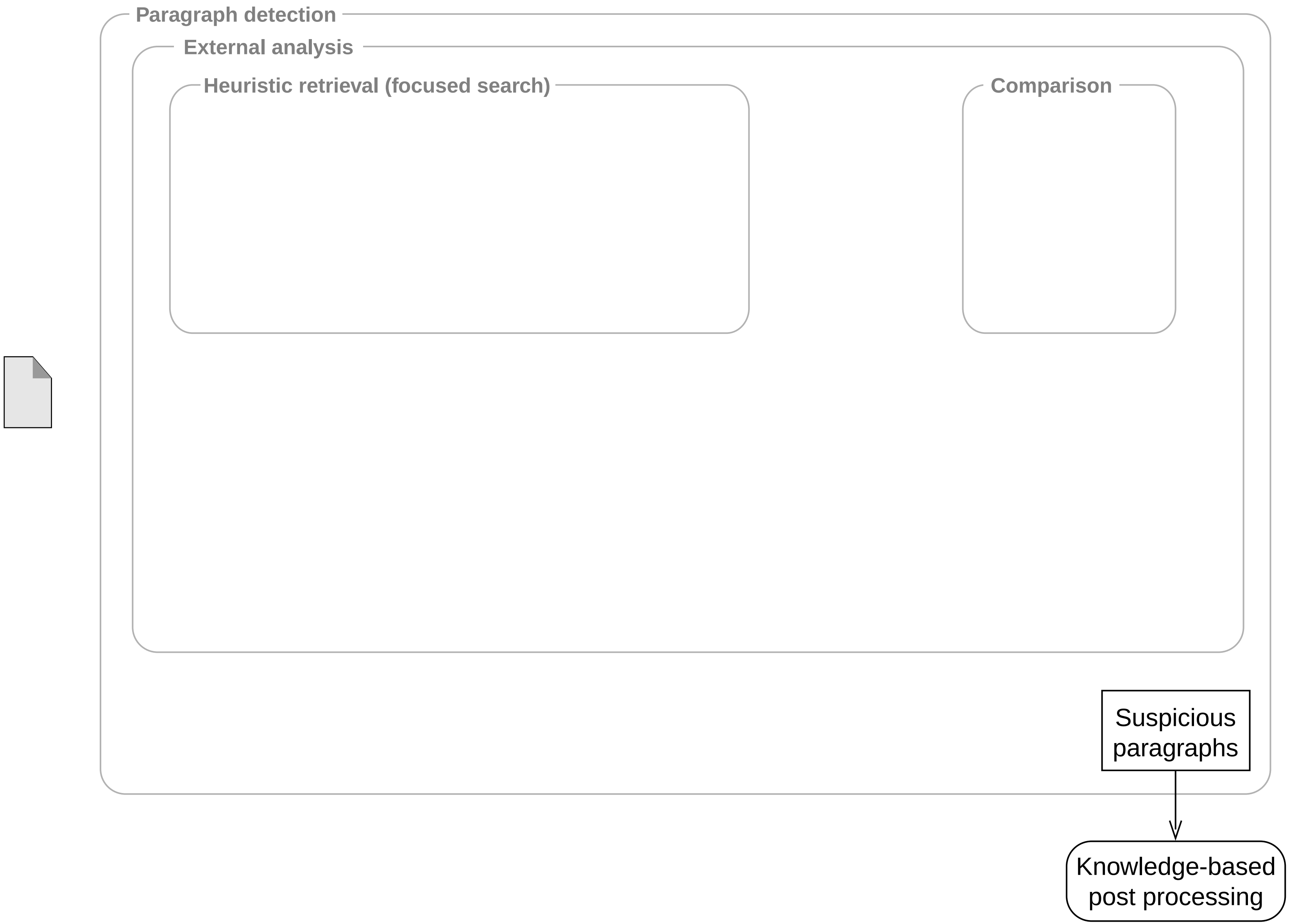
Overview

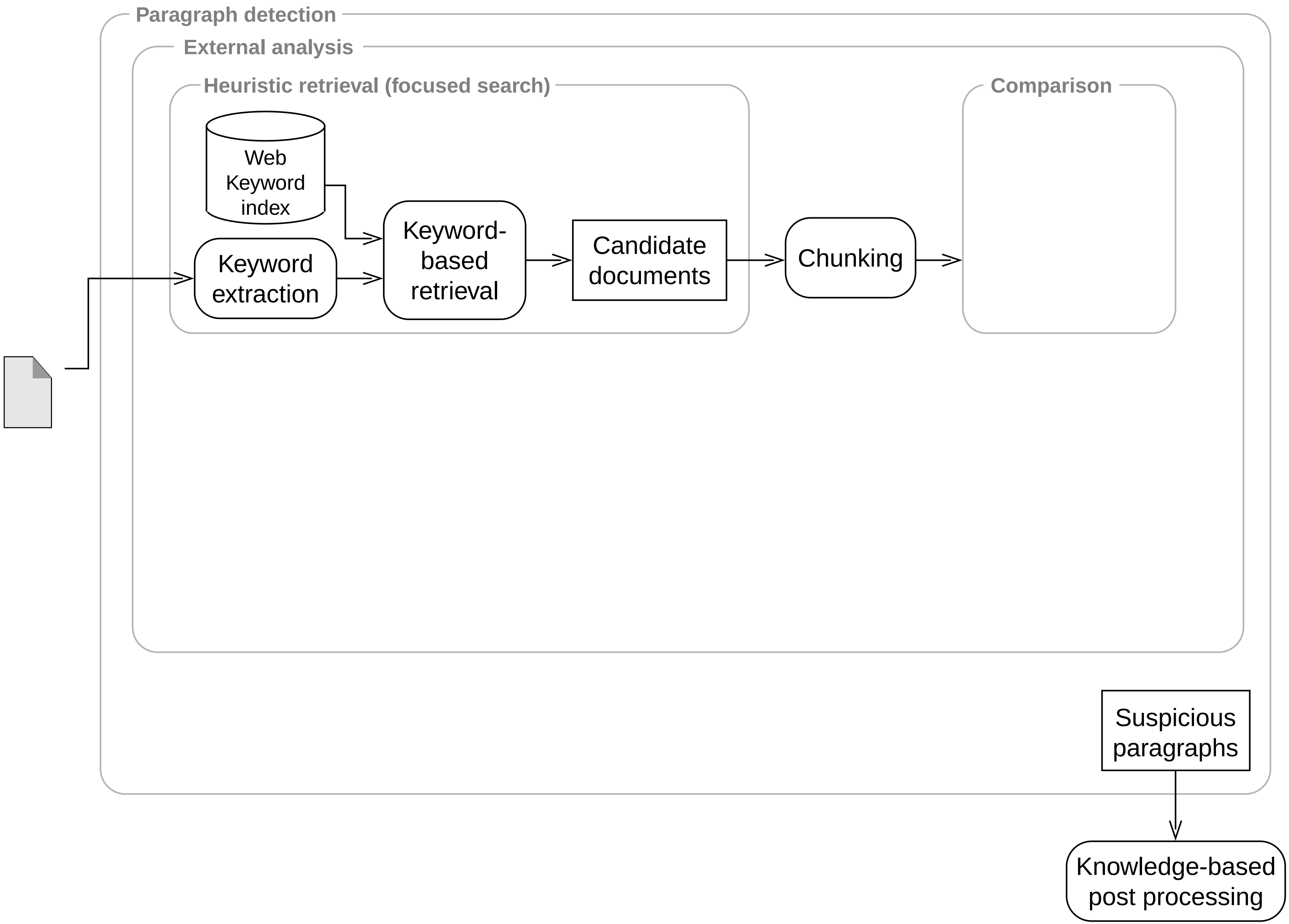
Taxonomy of Plagiarism Offenses

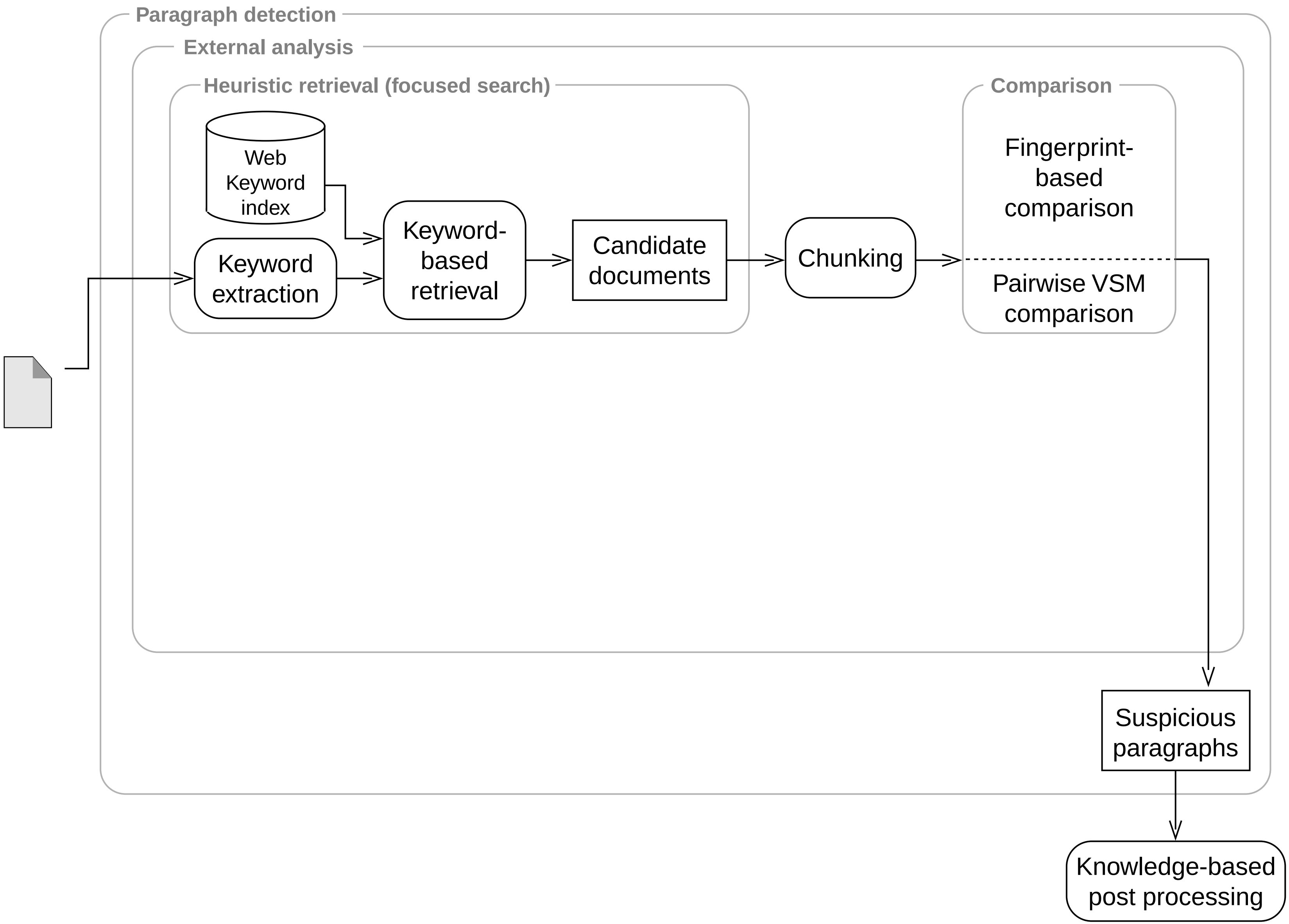


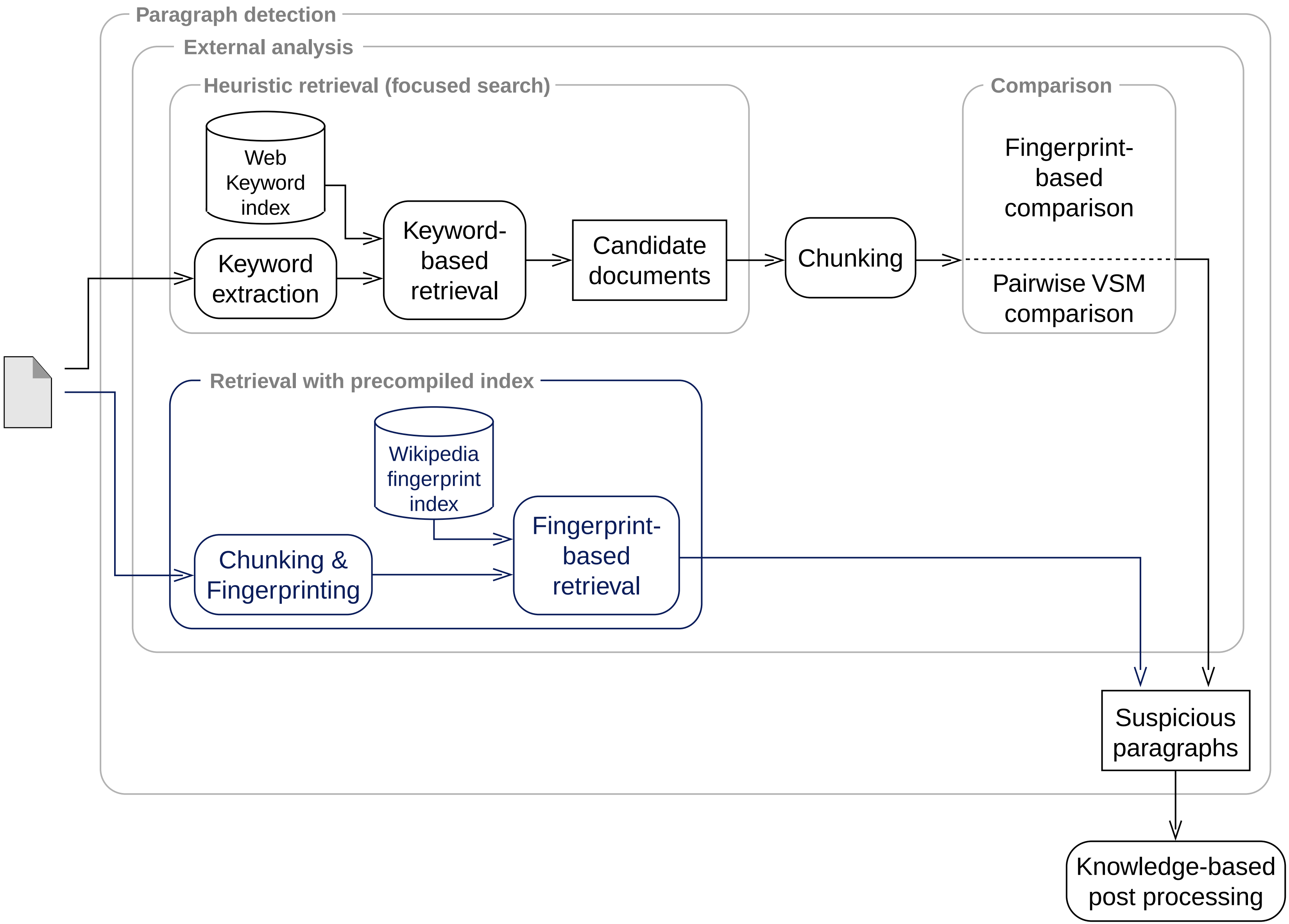
Paragraph detection

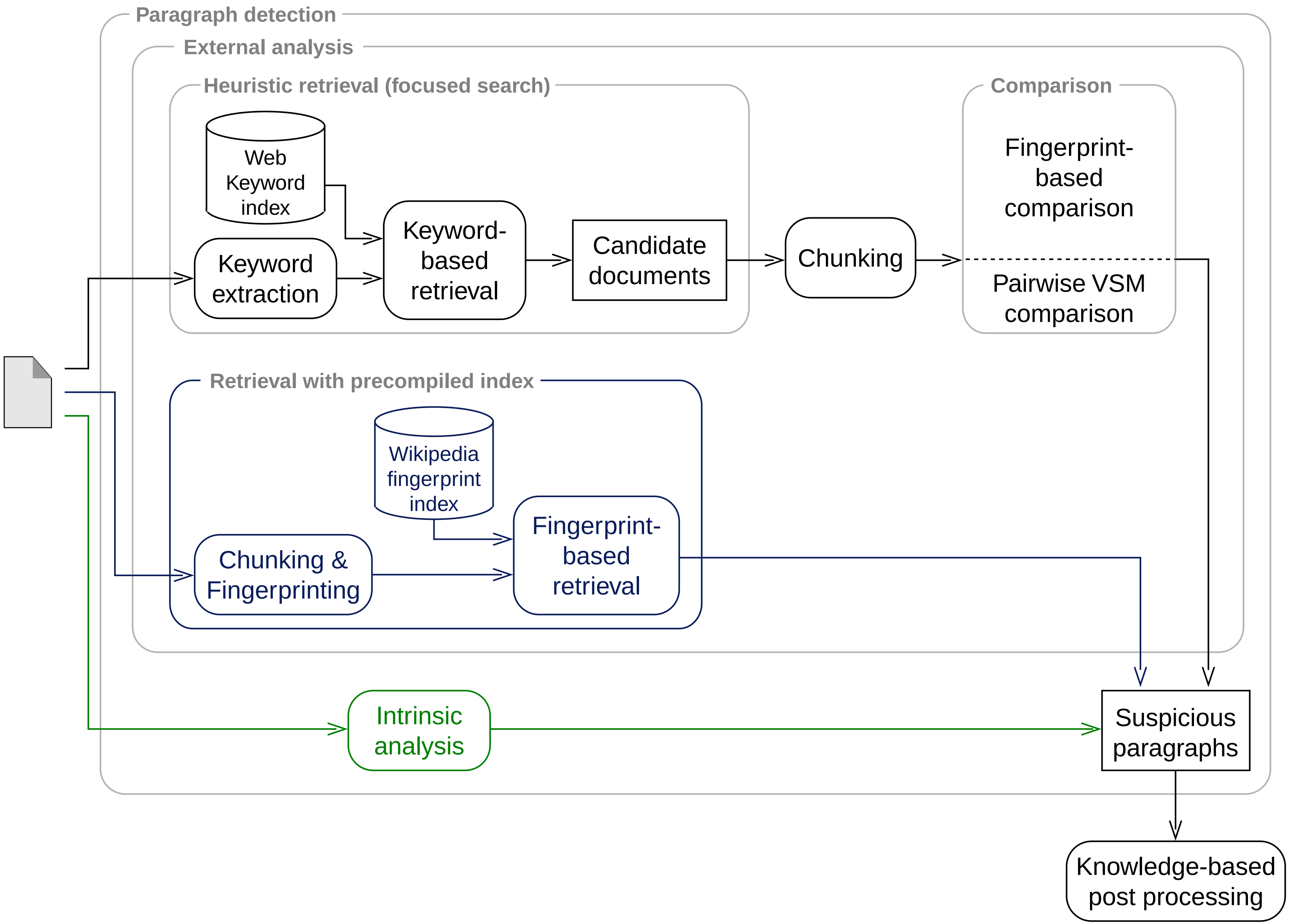
Knowledge-based post processing

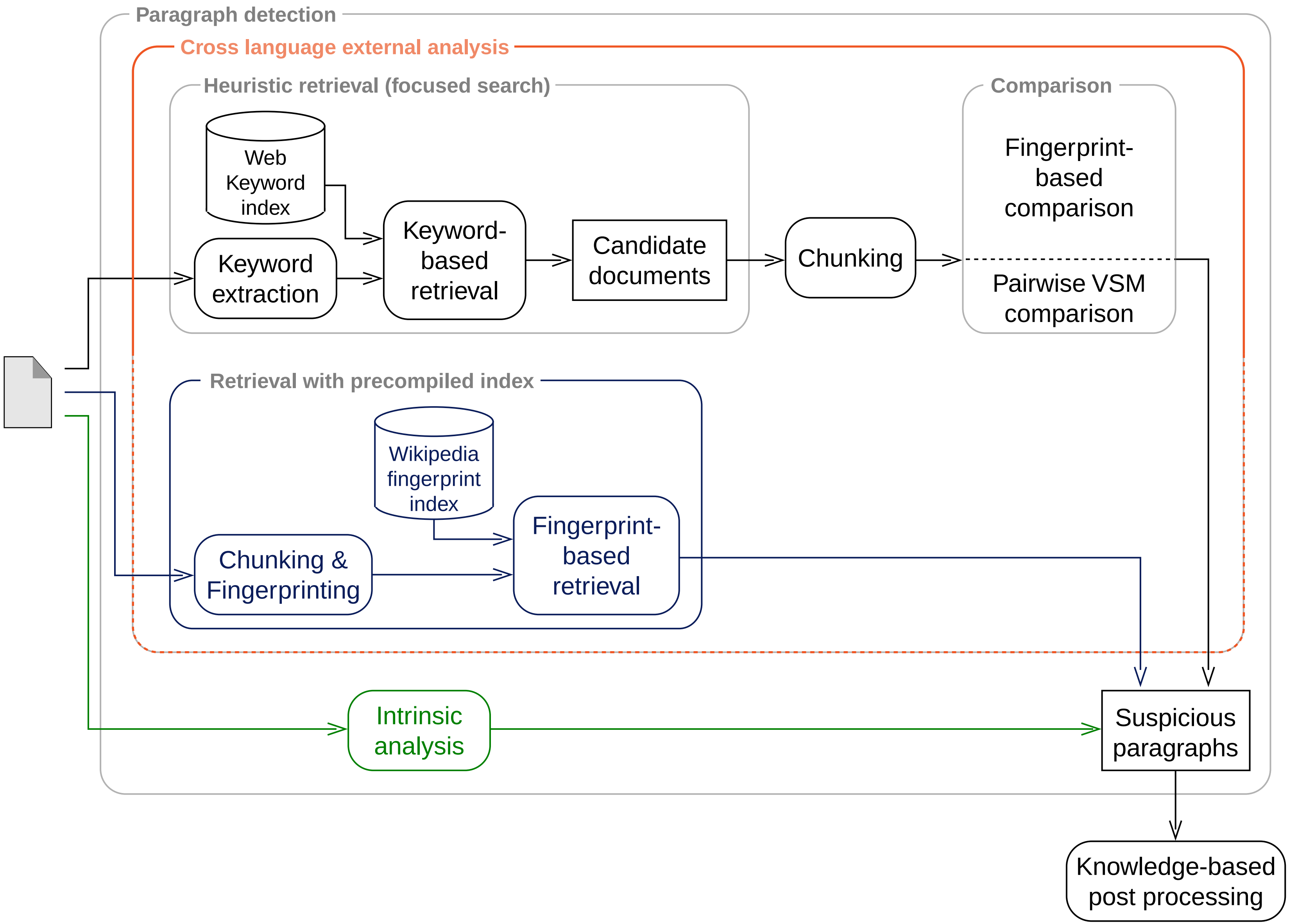

Paragraph detection
External analysis
Heuristic retrieval (focused search)
Comparison
Suspicious paragraphs
Knowledge-based post processing


Paragraph detection
External analysis
Heuristic retrieval (focused search)
Comparison
Web
Keyword
index
Keyword
extraction
Keyword-
based
retrieval
Candidate
documents
Chunking
Suspicious
paragraphs
Knowledge-based
post processing


Paragraph detection
External analysis
Heuristic retrieval (focused search)
Web
Keyword
index
Keyword extraction
Keyword-based retrieval
Candidate documents
Chunking
Comparison
Fingerprint-based comparison
Pairwise VSM comparison
Suspicious paragraphs
Knowledge-based post processing


Paragraph detection
External analysis
Heuristic retrieval (focused search)
Web Keyword index
Keyword extraction
Keyword-based retrieval
Candidate documents
Chunking
Comparison
Fingerprint-based comparison
Pairwise VSM comparison
Retrieval with precompiled index
Wikipedia fingerprint index
Chunking & Fingerprinting
Fingerprint-based retrieval
Suspicious paragraphs
Knowledge-based post processing


Paragraph detection
External analysis
Heuristic retrieval (focused search)
Web Keyword index
Keyword extraction
Keyword-based retrieval
Candidate documents
Chunking
Comparison
Fingerprint-based comparison
Pairwise VSM comparison
Retrieval with precompiled index
Wikipedia fingerprint index
Chunking & Fingerprinting
Fingerprint-based retrieval
Intrinsic analysis
Suspicious paragraphs
Knowledge-based post processing


Paragraph detection
Cross language external analysis
Heuristic retrieval (focused search)
Web Keyword index
Keyword extraction
Keyword-based retrieval
Candidate documents
Chunking
Comparison
Fingerprint-based comparison
Pairwise VSM comparison
Retrieval with precompiled index
Wikipedia fingerprint index
Chunking & Fingerprinting
Fingerprint-based retrieval
Intrinsic analysis
Suspicious paragraphs
Knowledge-based post processing

Overview

Examples for Identification Technology

- Level 1. Identity analysis for paragraphs.

  MD5 hashing

- Level 2. Synchronized identity analysis for paragraphs.

  hashed breakpoint chunking

- Level 3. Tolerant similarity analysis for paragraphs.

  Fuzzy-fingerprinting

- Level 4. Intrinsic (style) analysis without a reference corpus.

  statistical outlier analysis with Bayes, meta learning with logistic regression

- Level 5. Correct citation.

  knowledge-based analysis

Overview

Current research is corpus-centered, “external plagiarism analysis”.

[Brin et al. 1995, Monostori et al. 2001-2004, Stein et al. 2004-2006, etc.]

External plagiarism analysis formulated as decision problem:

Problem. AVEXTERN (AV stands for Authorship Verification)

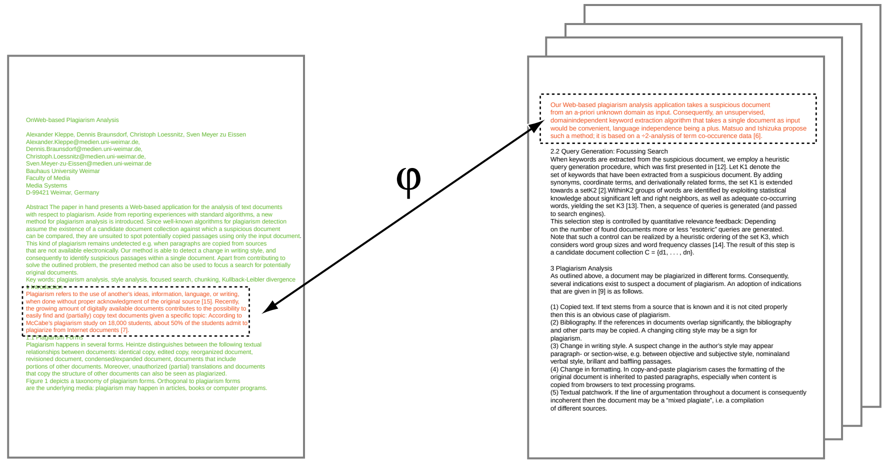
Given. A text d, allegedly written by author A, and set of texts D, $D = \{d_1, \ldots, d_n\}$, written by an arbitrary number of authors.

Question. Does d contain sections whose similarity to sections in D is above a threshold θ?

Overview

Basic Principle

- Partition each document in meaningful sections, also called “chunks”.
- Do a pairwise comparison using a similarity function φ.



suspicious document corpus documents

Complexity:

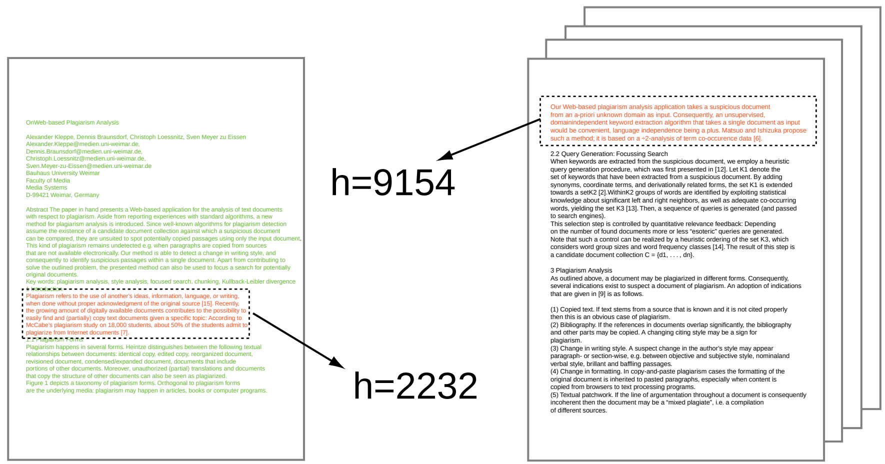
n documents in corpus, c chunks per document on average

➜ $O(n \cdot c^2)$ comparisons

Overview

Comparison with Fingerprints (Level 1)

- Partition each document into equidistant sections.
- Compute fingerprints of the chunks using a hash function h.
- Put all hashes into a hash table. A collision indicates matching chunks.



suspicious document corpus documents

Complexity:

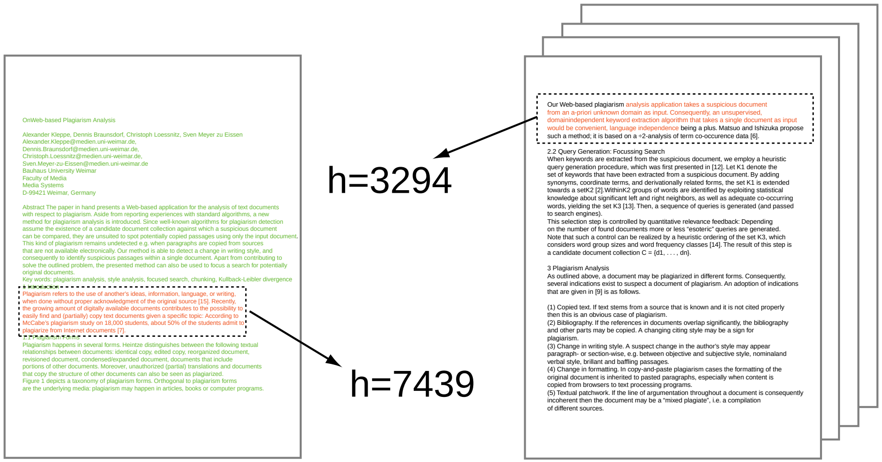
n documents in corpus, c chunks per document on average

➜ $O(n \cdot c)$ operations (fingerprint generation, hash table operations)

Overview

Comparison with Fingerprints (Level 2)

- Partition each document into *synchronized* sections.
- Compute fingerprints of the chunks using a hash function h.
- Put all hashes into a hash table. A collision indicates matching chunks.



suspicious document corpus documents

Complexity:

n documents in corpus, c chunks per document on average

→ $O(n \cdot c)$ operations (fingerprint generation, hash table operations)

Overview

Comparison with Fingerprints (Level 3)

Discussion:

- Hashing is fast, but sensitive to smallest changes:

$$h(c_1) = h(c_2) \Rightarrow c_1 = c_2$$ (with very high probability)

Current research:

- Focus on *fuzzy* hash functions h_φ:

$$h_\varphi(c_1) = h_\varphi(c_2) \quad \Rightarrow \quad P(\varphi(c_1, c_2) > \theta) \; \geq \; 1 - \varepsilon$$ [Stein 2005-07]

- Fuzzy hash functions allow for large chunk sizes (speed-up)
- Fuzzy hash functions are not sensitive to small changes

Plagiarism Corpus

Plagiarism Corpus

PAN Plagiarism Corpus 2009 (PAN-PC-09)

The PAN-PC-09 is a new large-scale resource for the controlled evaluation of plagiarism detection algorithms. [1]

Corpus overview:

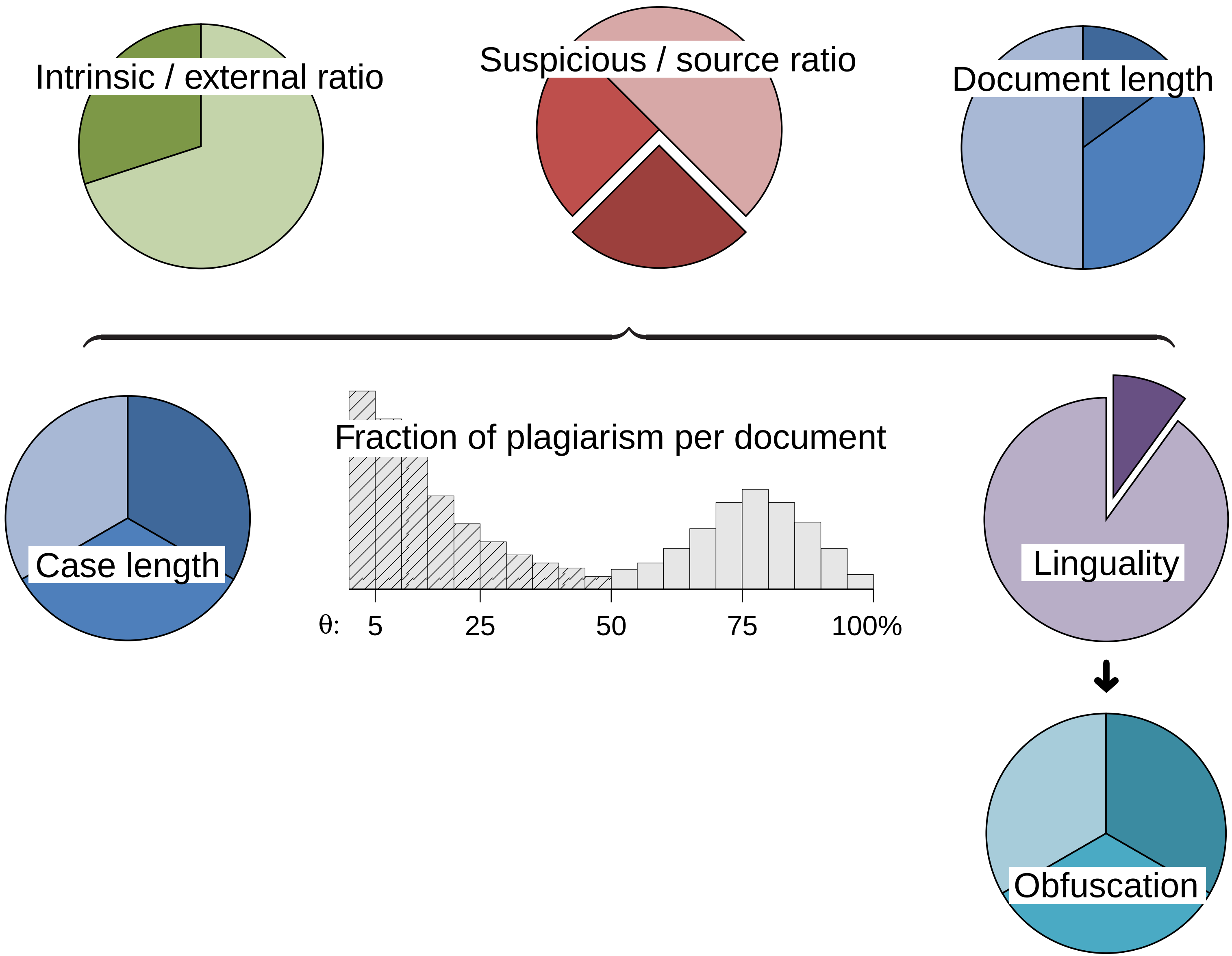
- 41 223 text documents (obtained from 22 874 books from the Project Gutenberg [2])
- 94 202 plagiarism cases
- 70% is dedicated to external plagiarism detection,
  30% is dedicated to intrinsic plagiarism detection

- Types of cases: monolingual with and without obfuscation, and cross-lingual
- Authenticity of cases: real, emulated, and artificial

[1] Webis at Bauhaus-Universität Weimar and NLEL at Universidad Politécnica de Valencia. PAN Plagiarism Corpus PAN-PC-09. `http://www.uni-weimar.de/medien/webis/research/corpora`, 2009. M. Potthast, A. Eiselt, B. Stein, A. Barrón-Cedeño, and P. Rosso (editors).

[2] `http://www.gutenberg.org`

Plagiarism Corpus



Plagiarism Corpus

Plagiarism Obfuscation Synthesis

Plagiarists often “modify” the text they plagiarize in order to obfuscate their offense.

- Obfuscation synthesis task:
  Given a section of text s_x, create a section s_q
  which has a high content similarity to s_x under some retrieval model
  but with a different word order or wording than s_x.

- Optimal obfuscation synthesizer:
  $s_x =$ “The quick brown fox jumps over the lazy dog.”

  $s_q^* =$ “Over the dog which is lazy jumps quickly the fox which is brown.”
  $s_q^* =$ “Dogs are lazy which is why brown foxes quickly jump over them.”
  $s_q^* =$ “A fast bay-colored vulpine hops over an idle canine.”

- Obfuscation Synthesis Strategies:
  (a) Random text operations
  (b) Semantic word variation
  (c) POS-preserving word shuffling

Plagiarism Corpus

Plagiarism Obfuscation Synthesis

Random text operations:
Given s_x, s_q is created by shuffling, removing, inserting, or replacing words or short phrases at random.

Examples:

$s_x =$ “The quick brown fox jumps over the lazy dog.”

$s_q =$ “over The. the quick lazy dog context jumps brown fox”
$s_q =$ “over jumps quick brown fox The lazy. the”
$s_q =$ “brown jumps the. quick dog The lazy fox over”

Plagiarism Corpus

Plagiarism Obfuscation Synthesis

Semantic word variation:
Given s_x, s_q is created by replacing each word by one of its synonyms, antonyms, hyponyms, or hypernyms, chosen at random.

Examples:

$s_x =$ “The quick brown fox jumps over the lazy dog.”

$s_q =$ “The quick brown dodger leaps over the lazy canine.”
$s_q =$ “The quick brown canine jumps over the lazy canine.”
$s_q =$ “The quick brown vixen leaps over the lazy puppy.”

Plagiarism Corpus

Plagiarism Obfuscation Synthesis

POS-preserving word shuffling:
Given s_x its sequence of parts of speech (POS) is determined. Then, s_q is created by shuffling words at random while the original POS sequence is maintained.

Examples:

$s_x =$ “The quick brown fox jumps over the lazy dog.”

POS $=$ “DT JJ JJ NN VBZ IN DT JJ NN .”

$s_q =$ “The brown lazy fox jumps over the quick dog.”
$s_q =$ “The lazy quick dog jumps over the brown fox.”
$s_q =$ “The brown lazy dog jumps over the quick fox.”

Plagiarism Corpus

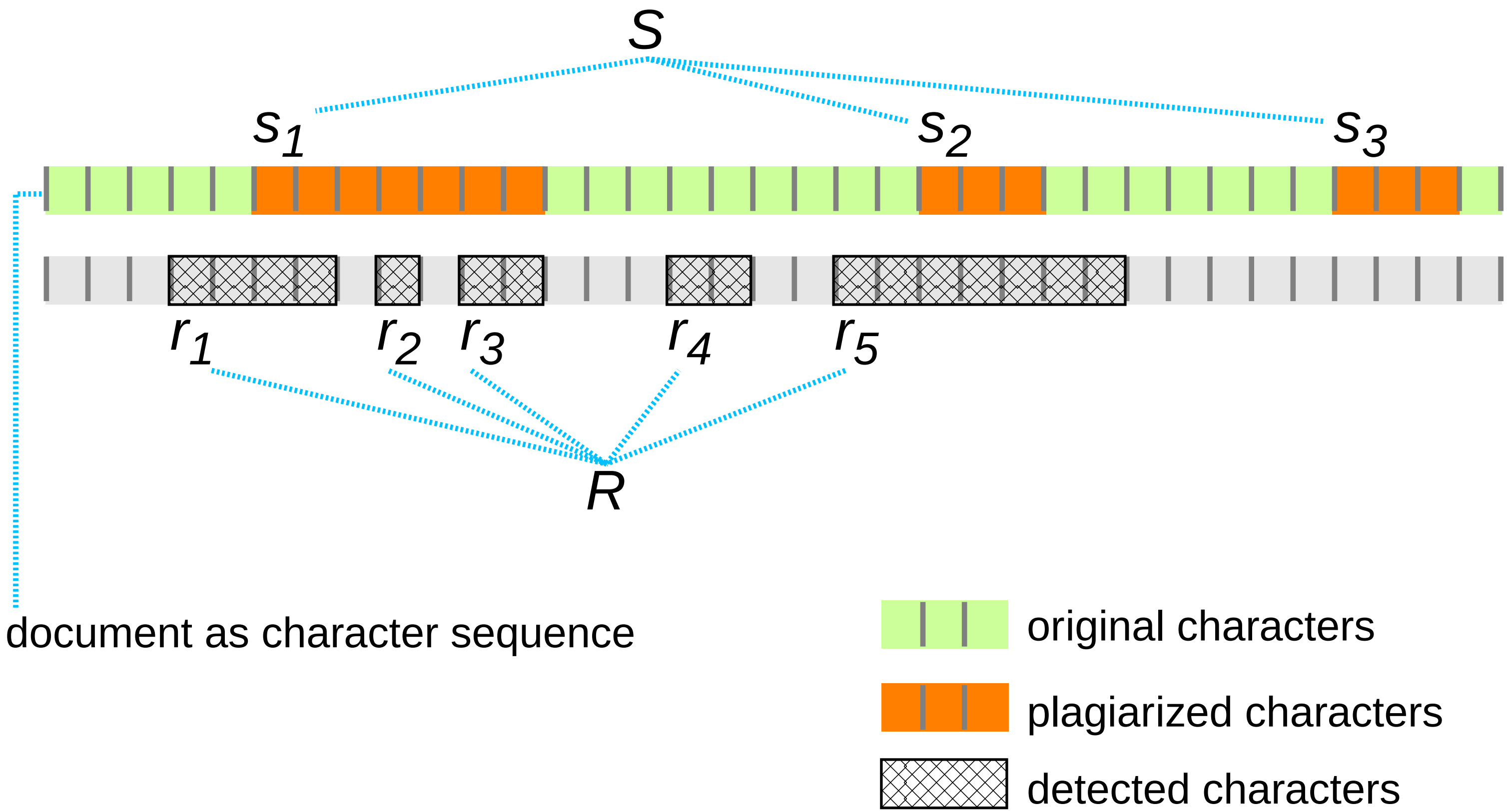
Critical Remarks

- Accidental similarities between suspicious and source documents.
- Anomalies in the plagiarized text produced by the obfuscation synthesizers.
- Inaccurate simulation of Web retrieval.

Detection Performance Measures

Detection Performance Measures

Terminology

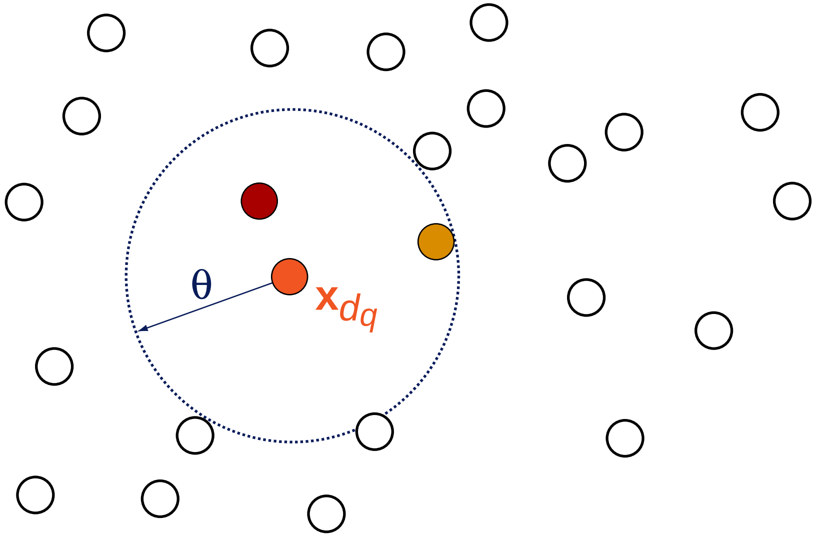


- $s_i \in S$ Plagiarized section from the set of all plagiarized sections.
- $r_i \in R$ Detected section from the set of all detected sections.

Hash-based Search: Motivation

Hash-based Search: Motivation

Nearest Neighbor Search

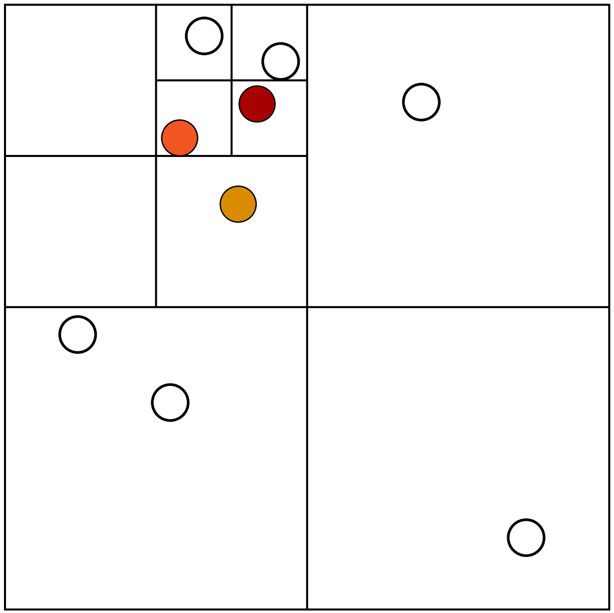


Applications:

- elimination of duplicates / near duplicates
- identification of versioned and plagiarized documents
- retrieval of similar documents
- identification of source code plagiarism

Hash-based Search: Motivation

Nearest Neighbor Search



Indexing with space partitioning methods:

- Quad-tree.

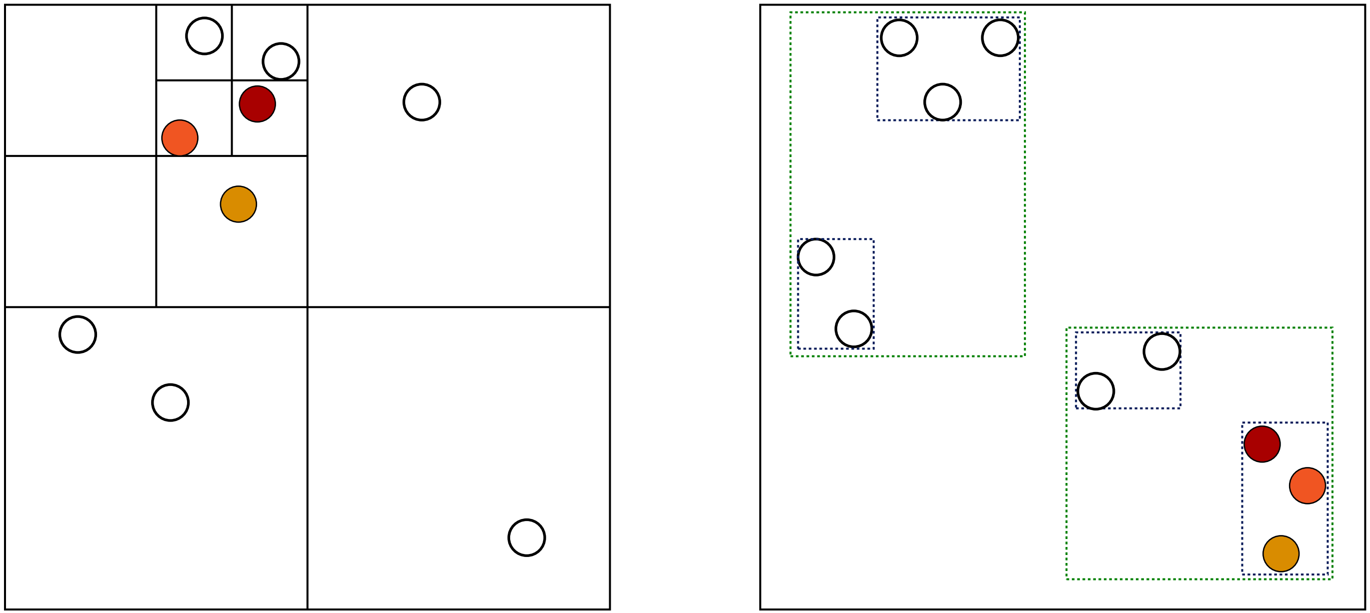
  Split the space recursively into sub-squares until only a few points left.

  Space exponential in dimension; time exponential in dimension.

- Kd-tree. Linear space; exponential query time is still possible.

Hash-based Search: Motivation

Nearest Neighbor Search



Indexing with data partitioning methods:

- R-tree.

  Bottom-up; heuristically construct minimum bounding regions for points

  Works well for low dimensions (< 10).

- Rf-tree, X-tree, . . .

Hash-based Search: Motivation

Document Representation and Search

The nearest neighbor problem cannot be solved efficiently in high dimensions by partitioning methods.

> *"Existing methods are outperformed on average by a simple sequential scan, if the number of dimensions exceeds around 10."*

[Weber 99, Gionis/Indyk/Motwani 99-04]

Hash-based Search: Motivation

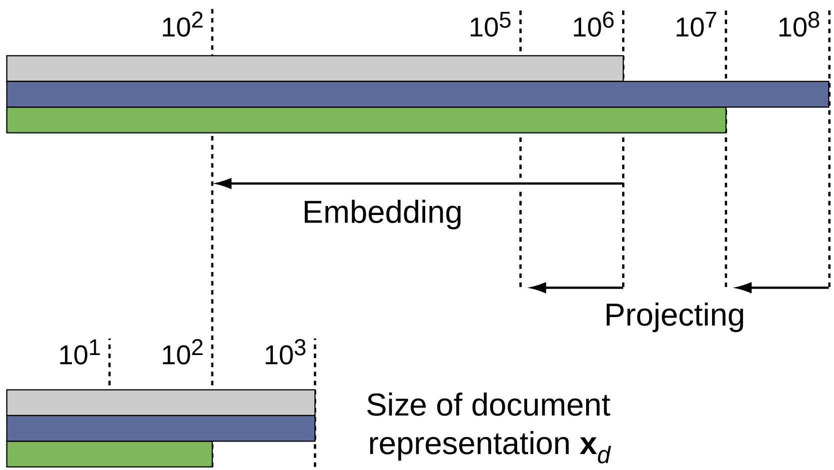
Document Representation and Search

The nearest neighbor problem cannot be solved efficiently in high dimensions by partitioning methods.

> *"Existing methods are outperformed on average by a simple sequential scan, if the number of dimensions exceeds around 10."*
>
> [Weber 99, Gionis/Indyk/Motwani 99-04]

English Wikipedia:

Dictionary	**Number of dimensions**
1-gram space	3 921 588
4-gram space	274 101 016
8-gram space	373 795 734
Shingling space	75 659 644



Hash-based Search: Motivation

Document Representation and Search

Given the representation $\mathbf{x}_{d_q}$ of a query document and a collection D.

- Linear comparison under some BOW representation
  ➜ Similarity ranking (baseline)

$$\begin{pmatrix} 0.02 \\ 0.0 \\ 0.01 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.02 \\ 0.07 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.1 \\ 0.2 \\ 0.0 \\ 0.1 \\ 0.2 \\ \vdots \\ 0.1 \\ 0.3 \\ 0.0 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.0 \\ 0.1 \\ 0.0 \\ 0.04 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.04 \\ 0.0 \\ 0.03 \end{pmatrix} \quad \ldots \quad \begin{pmatrix} 0.07 \\ 0.0 \\ 0.0 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.01 \\ 0.02 \\ 0.03 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.0 \\ 0.0 \\ 0.05 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.08 \\ 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.02 \\ 0.0 \\ 0.01 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.06 \\ 0.09 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.0 \\ 0.01 \\ 0.06 \\ 0.0 \\ 0.01 \\ \vdots \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.03 \end{pmatrix} \begin{pmatrix} 0.04 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.05 \\ \vdots \\ 0.01 \\ 0.02 \\ 0.03 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.0 \\ 0.0 \\ 0.01 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.02 \\ 0.06 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.1 \\ 0.0 \\ 0.09 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.05 \end{pmatrix}$$

Hash-based Search: Motivation

Document Representation and Search

Given the representation $\mathbf{x}_{d_q}$ of a query document and a collection D.

- Linear comparison under some BOW representation
  ➜ Similarity ranking (baseline)

- Linear comparison under some compact representation
  ➜ Acceptable similarity ranking (85% recall at $\varphi > 0.5$)

$$\begin{pmatrix} 0.02 \\ 0.0 \\ 0.01 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.02 \\ 0.07 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.1 \\ 0.2 \\ 0.0 \\ 0.1 \\ 0.2 \\ \vdots \\ 0.1 \\ 0.3 \\ 0.0 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.0 \\ 0.1 \\ 0.0 \\ 0.04 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.04 \\ 0.0 \\ 0.03 \end{pmatrix} \quad \ldots \quad \begin{pmatrix} 0.07 \\ 0.0 \\ 0.0 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.01 \\ 0.02 \\ 0.03 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.0 \\ 0.0 \\ 0.05 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.08 \\ 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.02 \\ 0.0 \\ 0.01 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.06 \\ 0.09 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.0 \\ 0.01 \\ 0.06 \\ 0.0 \\ 0.01 \\ \vdots \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.03 \end{pmatrix} \begin{pmatrix} 0.04 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.05 \\ \vdots \\ 0.01 \\ 0.02 \\ 0.03 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.0 \\ 0.0 \\ 0.01 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.02 \\ 0.06 \\ 0.0 \end{pmatrix} \begin{pmatrix} 0.1 \\ 0.0 \\ 0.09 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.05 \end{pmatrix}$$

Hash-based Search: Motivation

Document Representation and Search

Given the representation $\mathbf{x}_{d_q}$ of a query document and a collection D.

- Linear comparison under some BOW representation
  ➜ Similarity ranking (baseline)

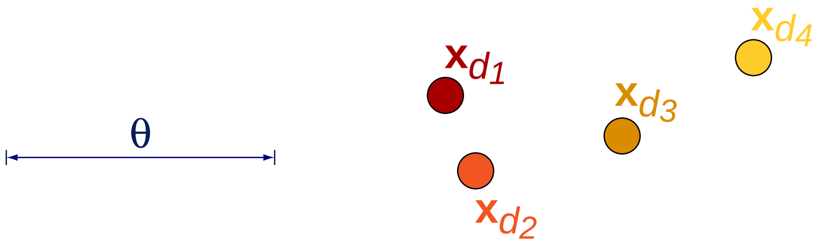
- Linear comparison under some compact representation
  ➜ Acceptable similarity ranking (85% recall at $\varphi > 0.5$)

- Comparison in constant time with a similarity-sensitive hash function h_φ
  ➜Binary decision wrt. threshold θ (similar if $\varphi > \theta$ / not similar if $\varphi \leq \theta$)

$$
\begin{array}{cccccccccccc}
\mathbf{124298} & \mathbf{456723} & \mathbf{546781} & & \mathbf{342509} & \mathbf{129842} & \mathbf{972653} & \mathbf{921345} & \mathbf{546719} & \mathbf{564214} & \mathbf{519461} \\
\begin{pmatrix} 0.02 \\ 0.0 \\ 0.01 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.02 \\ 0.07 \\ 0.0 \end{pmatrix} &
\begin{pmatrix} 0.1 \\ 0.2 \\ 0.0 \\ 0.1 \\ 0.2 \\ \vdots \\ 0.1 \\ 0.3 \\ 0.0 \\ 0.0 \end{pmatrix} &
\begin{pmatrix} 0.0 \\ 0.1 \\ 0.0 \\ 0.04 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.04 \\ 0.0 \\ 0.03 \end{pmatrix} &
\cdots &
\begin{pmatrix} 0.07 \\ 0.0 \\ 0.0 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.01 \\ 0.02 \\ 0.03 \\ 0.0 \end{pmatrix} &
\begin{pmatrix} 0.0 \\ 0.0 \\ 0.05 \\ 0.1 \\ 0.0 \\ \vdots \\ 0.08 \\ 0.0 \\ 0.0 \\ 0.0 \end{pmatrix} &
\begin{pmatrix} 0.02 \\ 0.0 \\ 0.01 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.06 \\ 0.09 \\ 0.0 \end{pmatrix} &
\begin{pmatrix} 0.0 \\ 0.01 \\ 0.06 \\ 0.0 \\ 0.01 \\ \vdots \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.03 \end{pmatrix} &
\begin{pmatrix} 0.04 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.05 \\ \vdots \\ 0.01 \\ 0.02 \\ 0.03 \\ 0.0 \end{pmatrix} &
\begin{pmatrix} 0.0 \\ 0.0 \\ 0.01 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.02 \\ 0.06 \\ 0.0 \end{pmatrix} &
\begin{pmatrix} 0.1 \\ 0.0 \\ 0.09 \\ 0.0 \\ 0.0 \\ \vdots \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.05 \end{pmatrix}
\end{array}
$$

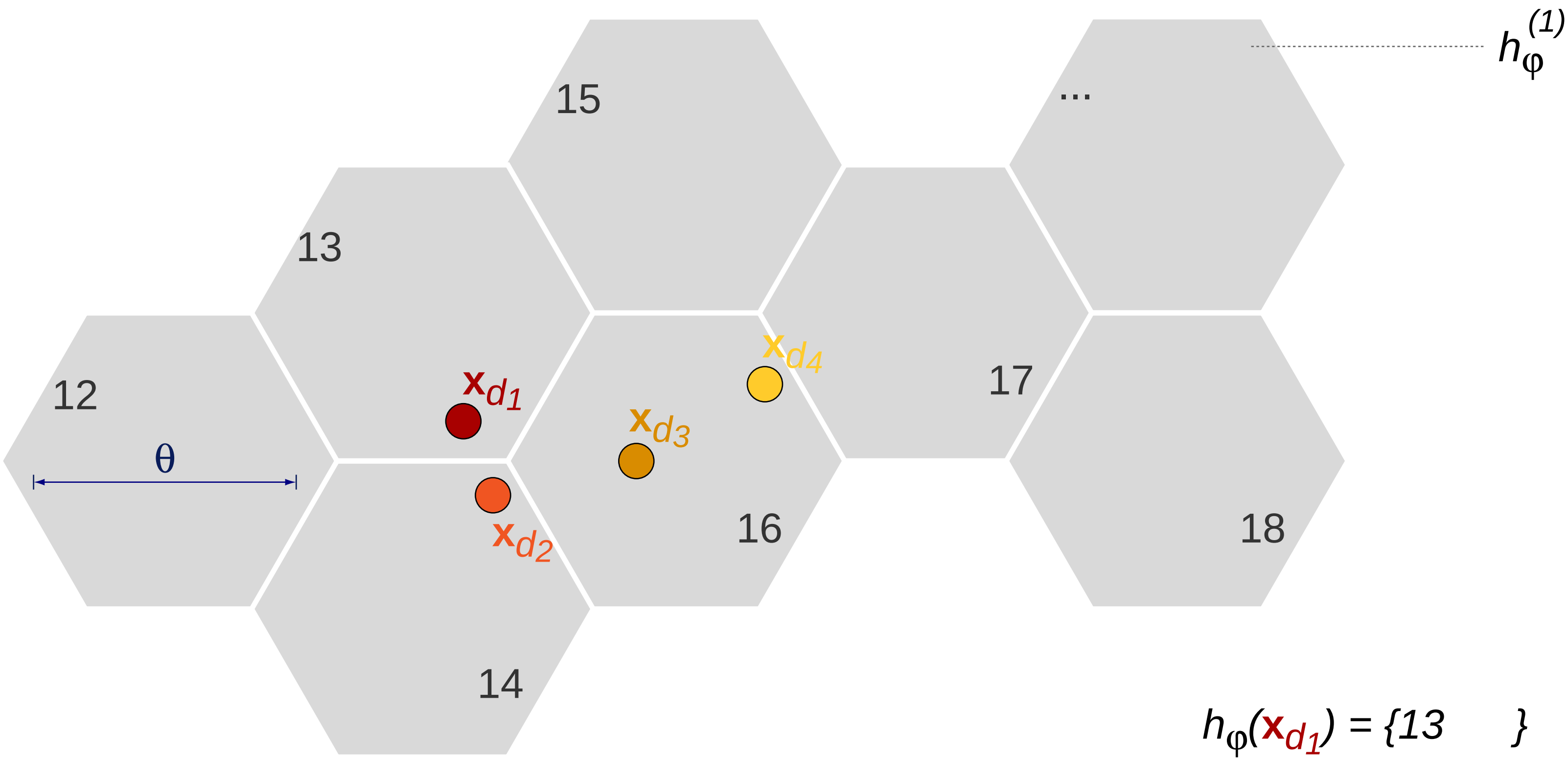
Hash-based Search: Motivation

Hash-based Search is a Space Partitioning Method



Hash-based Search: Motivation

Hash-based Search is a Space Partitioning Method



$h_\varphi(\mathbf{x}_{d_1}) = \{13 \quad\}$

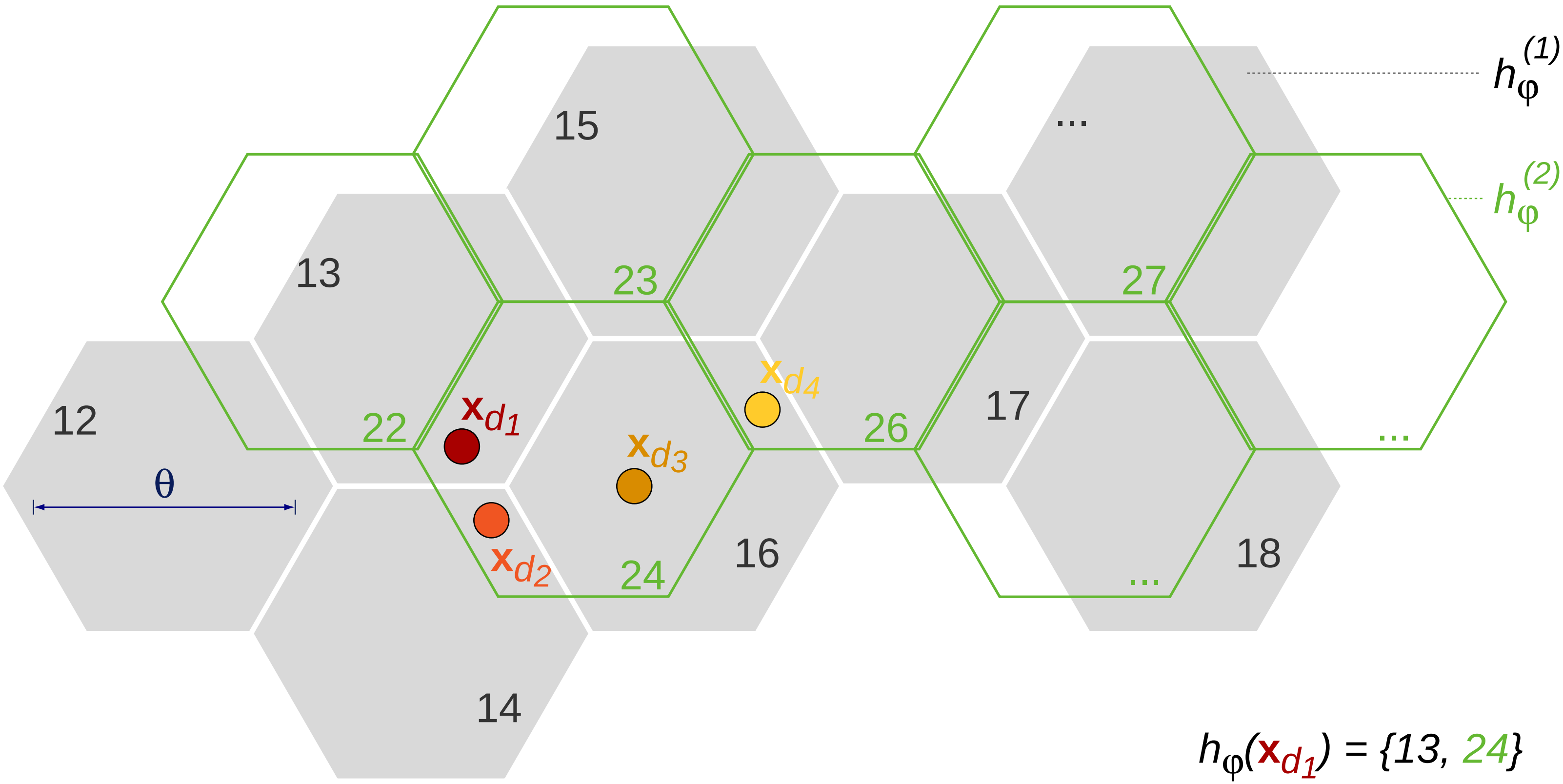
$h_\varphi(\mathbf{x}_{d_2}) = \{14 \quad\}$

$h_\varphi(\mathbf{x}_{d_3}) = \{16 \quad\}$

$h_\varphi(\mathbf{x}_{d_4}) = \{16 \quad\}$

Hash-based Search: Motivation

Hash-based Search is a Space Partitioning Method



$h_\varphi(\mathbf{x}_{d_1}) = \{13, 24\}$

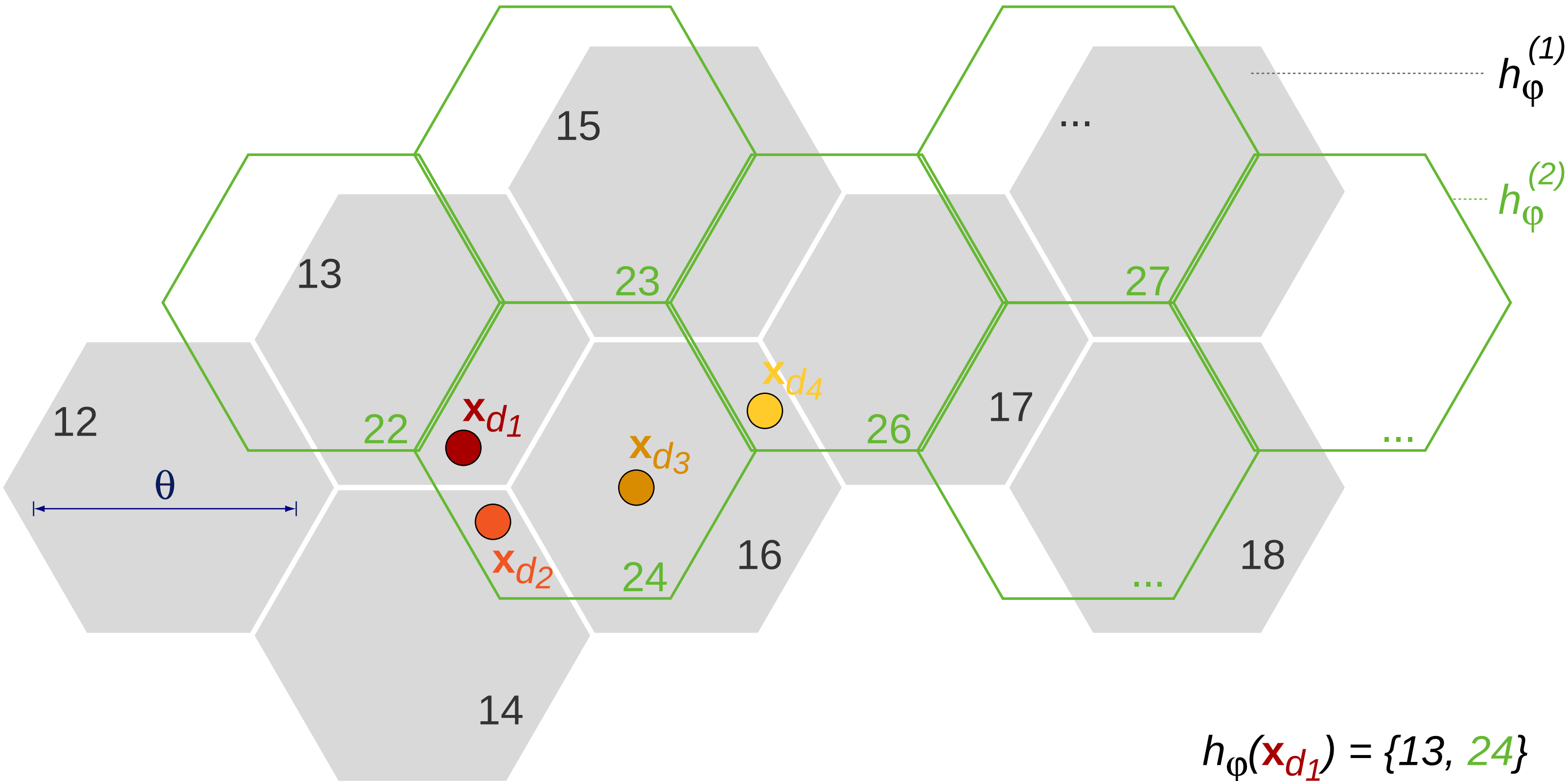
$h_\varphi(\mathbf{x}_{d_2}) = \{14, 24\}$

$h_\varphi(\mathbf{x}_{d_3}) = \{16, 24\}$

$h_\varphi(\mathbf{x}_{d_4}) = \{16, 26\}$

Hash-based Search: Motivation

Hash-based Search is a Space Partitioning Method



$h_\varphi(\mathbf{x}_{d_1})$ = {13, 24}

$h_\varphi(\mathbf{x}_{d_2})$ = {14, 24}

$h_\varphi(\mathbf{x}_{d_3})$ = {16, 24}

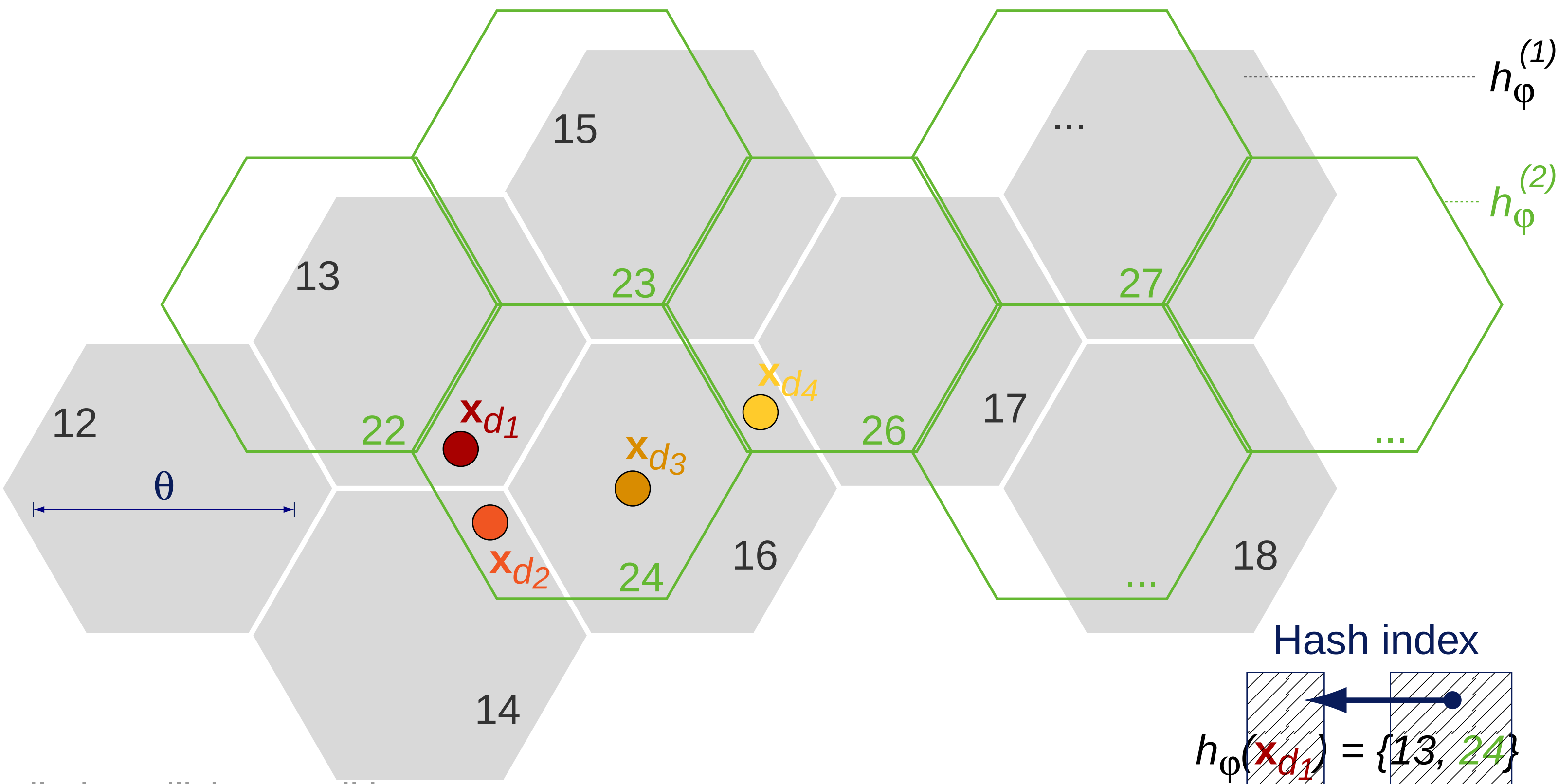
$h_\varphi(\mathbf{x}_{d_4})$ = {16, 26}

Similarity collision condition:

$$\boxed{\left(\ h^*_\varphi(\mathbf{x}_{d_1}) \cap h^*_\varphi(\mathbf{x}_{d_2})\ \right) \neq \emptyset \quad \Leftrightarrow \quad \varphi(\mathbf{x}_{d_1}, \mathbf{x}_{d_2}) > \theta}$$

Hash-based Search: Motivation

Hash-based Search is a Space Partitioning Method



$h_\varphi(\mathbf{x}_{d_1}) = \{13, 24\}$

$h_\varphi(\mathbf{x}_{d_2}) = \{14, 24\}$

$h_\varphi(\mathbf{x}_{d_3}) = \{16, 24\}$

$h_\varphi(\mathbf{x}_{d_4}) = \{16, 26\}$

Similarity collision condition:

$$\left(\ h^*_\varphi(\mathbf{x}_{d_1}) \cap h^*_\varphi(\mathbf{x}_{d_2})\ \right) \neq \emptyset \quad \Leftrightarrow \quad \varphi(\mathbf{x}_{d_1}, \mathbf{x}_{d_2}) > \theta$$

Hash-based Search: Motivation

Issues about Hash-based Search

- Hash-based search reduces a cont. similarity relation to a binary relation.
- Hash-based search is a space partitioning method.
- Space partitioning is realized by a similarity-sensitive hash function h_φ.

- Equal codes under h_φ indicate similar objects with a high probability.

  Precision: $h_\varphi(\mathbf{x}_{d_1}) \cap h_\varphi(\mathbf{x}_{d_2}) \neq \emptyset \quad \Rightarrow \quad P(\varphi(\mathbf{x}_{d_1}, \mathbf{x}_{d_2}) > \theta)$ is high

- h_φ maps similar objects on equal codes with a high probability.

  Recall: $\varphi(\mathbf{x}_{d_1}, \mathbf{x}_{d_2}) > \theta \quad \Rightarrow \quad P(h_\varphi(\mathbf{x}_{d_1}) \cap h_\varphi(\mathbf{x}_{d_2}) \neq \emptyset)$ is high

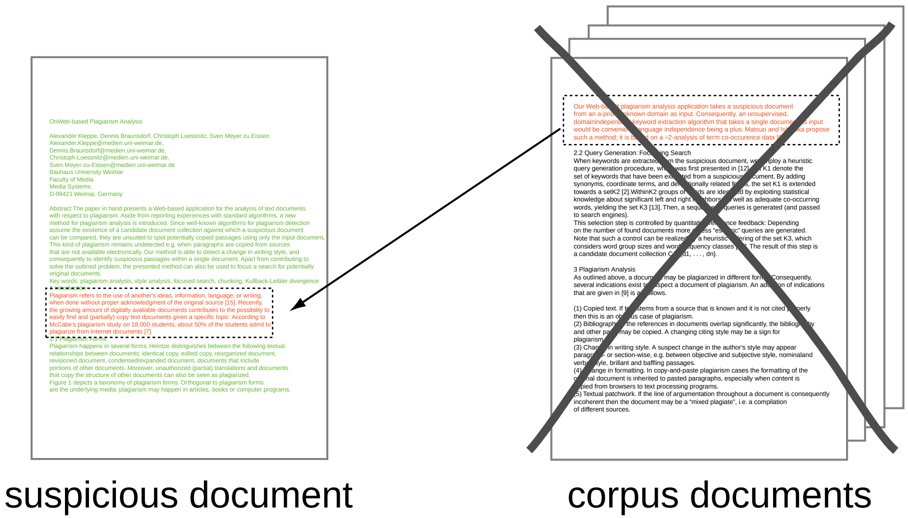
- h_φ must be multi-valued if D is partly unknown.
- A perfectly similarity-sensitive hash function h_φ^* may exist for each D.

Intrinsic Analysis and Authorship Verification

Intrinsic Analysis and Authorship Verification

Problem Setting

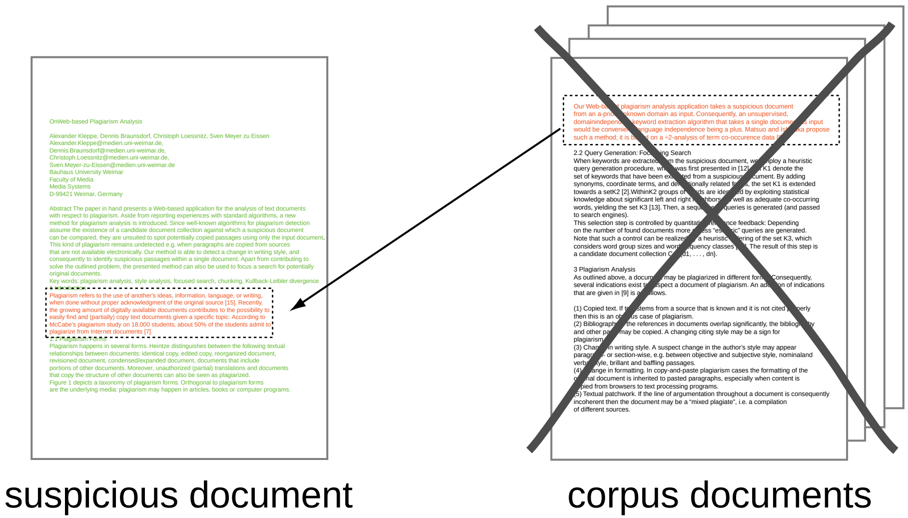
How to find a plagiarized section / foreign authorship without a reference corpus?



Intrinsic Analysis and Authorship Verification

Problem Setting

How to find a plagiarized section / foreign authorship without a reference corpus?



Formulated as decision problem:

Problem. AVFIND
Given. A text d, allegedly written by author A.
Question. Does d contain sections written by an author B, $B \neq A$?

Intrinsic plagiarism analysis and authorship verification (AV) are two sides of the same coin.

Intrinsic Analysis and Authorship Verification

Building Blocks for Authorship Verification

Pre-analysis			Classification	Post-processing	
Impurity assessment	Decomposition strategy	**Style model construction**	**Style outlier identification**	Improvement at section level	**Improvement at document level**
Document length analysis	Uniform length	Formatting	Two-class discriminant analysis	Citation analysis	Confidence-based majority decision
Genre Analysis	Structural boundaries	Surface analysis	One-class classifier: density estimation		Unmasking
Analysis of issuing institution	Text element boundaries	Structure analysis	One-class classifier: boundary estimation		Batch means
	Topical boundaries	Complexity measures	One-class classifier: reconstruction		Human inspection
		n-gram analysis			
		Language modeling			
		Dialectic analysis			

Intrinsic Analysis and Authorship Verification

Style Model Construction: Starting Points

Selected quantifiable feature classes (from easy to difficult):

- surface features
- structure and organization
- complexity measures
  - readability
  - writing complexity
  - vocabulary richness, diction
- dialectic power
  - argumentation consistency
  - argumentation strategy

For a machine-based identification, features have to be developed and operationalized within a style model $\mathcal{R}$.

Style Model Construction: Starting Points

Feature type	Stylometric feature	Unit of measure
surface	average paragraph length	paragraph
	average sentence length	sentence
	average word length	word
	average stop word portion	word
	spelling errors	word
	. . .	. . .
readability	Flesch Reading Ease Index	sentence, word
	Flesch Kincaid Grade Level	sentence, word
	Gunning Fog Index	sentence, word
	Dale Chall Index	sentence, word
writing complexity,	Honoré's R	word
vocabulary richness	Yule's K	word
	Kullback Leibler Divergence	word
	Word Frequency Class	word

Intrinsic Analysis and Authorship Verification

Style Model Construction: Word Frequency Class

Sentence 1: *“The values of the features are different.”*
Sentence 2: *“The feature’s values diverge.”*

Differences:

- “of the” vs. genitive-s ➜ part-of-speech analysis
  (average # prepositions, average # articles...)
- “are different” vs. “diverge” ➜ word frequency analysis

Intrinsic Analysis and Authorship Verification

Style Model Construction: Word Frequency Class

Let $\mathcal{C}$ be a (large) corpus of documents, and let

$f(w)$ denote the frequency, and

$r(w)$ denote the rank

of a word w in $\mathcal{C}$.

Zipf's Law: $f(w) \cdot r(w) =$ *constant*

$f(w_1) \cdot 1 \simeq f(w_2) \cdot 2 \simeq f(w_4) \cdot 4 \ldots$ (w_i ordered by rank)

Intrinsic Analysis and Authorship Verification

Style Model Construction: Word Frequency Class

Let $\mathcal{C}$ be a (large) corpus of documents, and let

$f(w)$ denote the frequency, and

$r(w)$ denote the rank

of a word w in $\mathcal{C}$.

Zipf's Law: $f(w) \cdot r(w) =$ *constant*

$f(w_1) \cdot 1 \simeq f(w_2) \cdot 2 \simeq f(w_4) \cdot 4 \ldots$ (w_i ordered by rank)

The word frequency class $\gamma(w)$ is defined as k if

$$2^{k-1} \leq \frac{f(w_1)}{f(w)} < 2^k$$

Examples: γ(different)=7, γ(diverge)=16

Averaging γ over a text d will quantify d's word customariness.

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams   ($n = 4$)

Example:

Our Web-based plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams   ($n = 4$)

Example:

$\boxed{\text{Our W}}$eb-based plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams ($n = 4$)

Example:

Our Web-based plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams ($n = 4$)

Example:

Our $\boxed{\text{r Web}}$-based plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams   ($n = 4$)

Example:

Our $\boxed{\text{Web-b}}$ased plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams $(n = 4)$
- word n-grams $(n = 3)$

Example:

$\boxed{\text{Our Web-based}}$ plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams $(n = 4)$
- word n-grams $(n = 3)$

Example:

Our $\boxed{\text{Web-based plagiarism}}$ analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams $(n = 4)$
- word n-grams $(n = 3)$

Example:

Our Web-based plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams $(n = 4)$
- word n-grams $(n = 3)$

Example:

Our Web-based plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams $(n = 4)$
- word n-grams $(n = 3)$
- part-of-speech n-grams $(n = 2)$

Example:

```
<pp> <a>              <n>          <n>         <n>          <v>    <det>
Our Web-based plagiarism analysis application takes the suspicious docu...
```

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams $(n = 4)$
- word n-grams $(n = 3)$
- part-of-speech n-grams $(n = 2)$

Example:

<pp> <a> <n> <n> <n> <v> <det>
Our Web-based plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams $(n = 4)$
- word n-grams $(n = 3)$
- part-of-speech n-grams $(n = 2)$

Example:

<pp> <a> $\boxed{\text{<n> <n>}}$ <n> <v> <det>
Our Web-based plagiarism analysis application takes the suspicious docu...

Intrinsic Analysis and Authorship Verification

Style Model Construction: n-Grams

Underlying alphabet for feature computation:

- character n-grams $(n = 4)$
- word n-grams $(n = 3)$
- part-of-speech n-grams $(n = 2)$

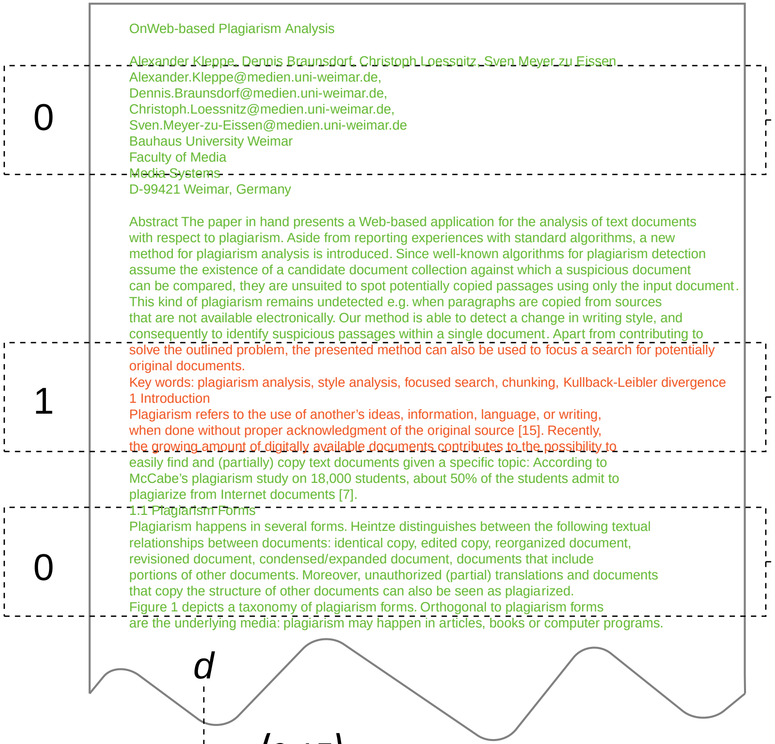
Example:

```
<pp>  <a>           <n>         <n>        <n>          <v>   <det>
Our Web-based plagiarism analysis application takes the suspicious docu...
```

Intrinsic Analysis and Authorship Verification

Style Model Construction: Language Modeling

Style Outlier Identification



$$\mathbf{s} = \begin{pmatrix} 0.16 \\ 0.20 \\ 0.65 \end{pmatrix} \qquad \mathbf{s}_\Delta = \begin{pmatrix} 0.014 \\ -0.008 \\ -0.050 \end{pmatrix}$$

$$\mathbf{s} = \begin{pmatrix} 0.24 \\ 0.09 \\ 0.54 \end{pmatrix} \qquad \mathbf{s}_\Delta = \begin{pmatrix} 0.090 \\ -0.121 \\ -0.160 \end{pmatrix}$$

$$\mathbf{s} = \begin{pmatrix} 0.14 \\ 0.20 \\ 0.70 \end{pmatrix} \qquad \mathbf{s}_\Delta = \begin{pmatrix} -0.051 \\ -0.011 \\ 0.000 \end{pmatrix}$$

$$\mathbf{d} = \begin{pmatrix} 0.15 \\ 0.21 \\ 0.70 \end{pmatrix}$$

Supervised learning situation: given are sections s_i from both the target class (author A), where $c(s) = 0$, and the outlier class (other authors), where $c(s) = 1$.

Intrinsic Analysis and Authorship Verification

Style Outlier Identification

Compute for each section the relative differences between section-specific style feature values and document-specific style feature values.

1. Let $\sigma_1, \ldots, \sigma_m$ denote style feature functions.

2. For each section $s \subseteq d$:

   - compute style model $\mathbf{s} = \begin{pmatrix} \sigma_1(s) \\ \vdots \\ \sigma_m(s) \end{pmatrix} \in \mathbf{R}^m$

   - compute relative deviations $\mathbf{s}_\Delta = \begin{pmatrix} \frac{\sigma_1(s) - \sigma_1(d)}{\sigma_1(d)} \\ \vdots \\ \frac{\sigma_m(s) - \sigma_m(d)}{\sigma_m(d)} \end{pmatrix} \in \mathbf{R}^m$

3. Learn an outlier hypothesis h from a sample $\{(\mathbf{s}_\Delta, c(s))\},\ c(s) \in \{0, 1\}$.

Intrinsic Analysis and Authorship Verification

Evaluation: Test Corpus

No benchmark corpus available. Our construction:

100 Documents from the ACM DL, each one “plagiarized”

- by hand,
- with up to 20% of text from other authors,
- in up to 5 different locations in each document.

XML template document:

```
<document url="http://...">
  ...original text...
    <plagiarized source="http://..."type=copied">
    ...plagiarized text...
    </plagiarized>
  ...original text...
</document>
```

→ 2^k instance documents for k “plagiarized” parts.

Intrinsic Analysis and Authorship Verification

Evaluation: Style Model Performance

Feature set:

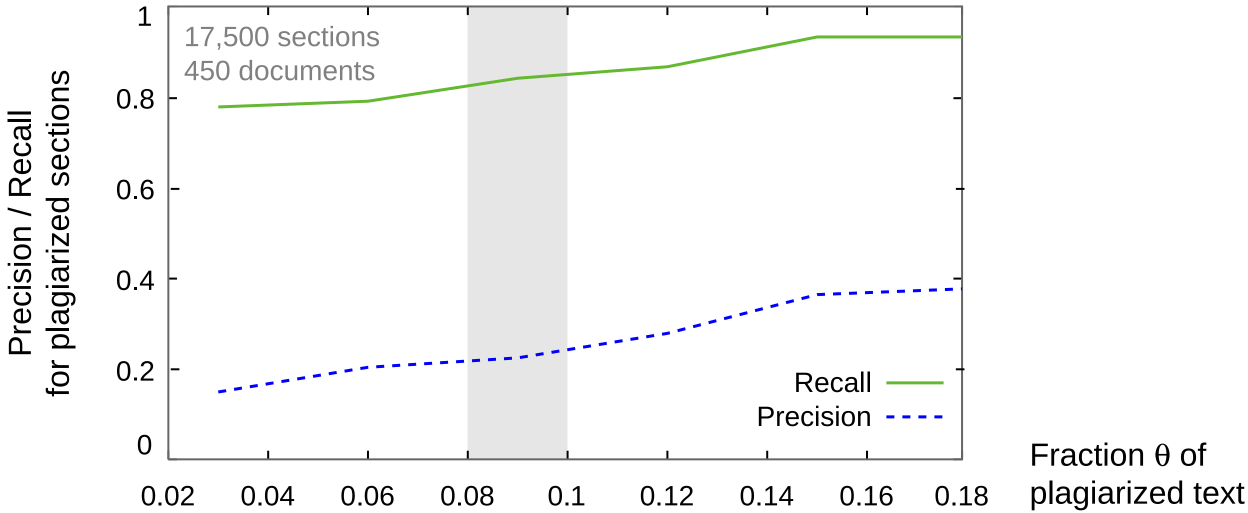
- 18 part-of-speech features
- average word frequency class
- average syllables per word
- average sentence length
- Gunning-Fog Index
- Flesch Readability Index

Results of a discriminant analysis on $\{(\mathbf{s}_\Delta, h(\mathbf{s}_\Delta))\}$ on our corpus:

- fraction θ of plagiarized sections is from [0.03; 0.18]
- about 30% precision and 85% recall for plagiarized sections
- the learning algorithm is not informed about the true value of θ

Intrinsic Analysis and Authorship Verification

Evaluation: Style Model Performance

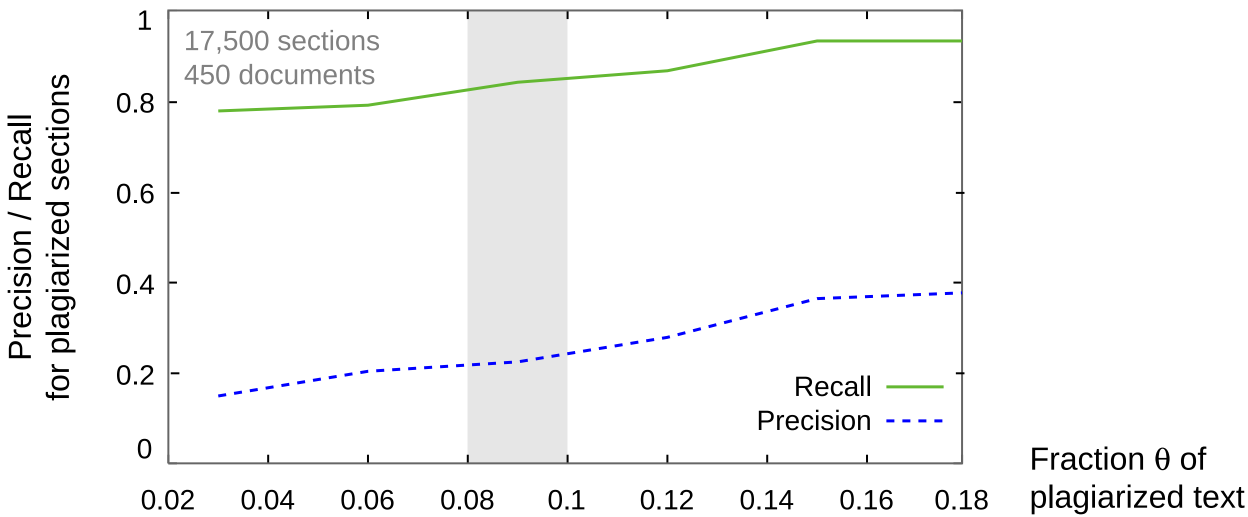


The unsatisfying precision is rooted in the class imbalance.

The Gretchenfrage: Are parts of d plagiarized, if we find an outlier?

Intrinsic Analysis and Authorship Verification

Evaluation: Style Model Performance

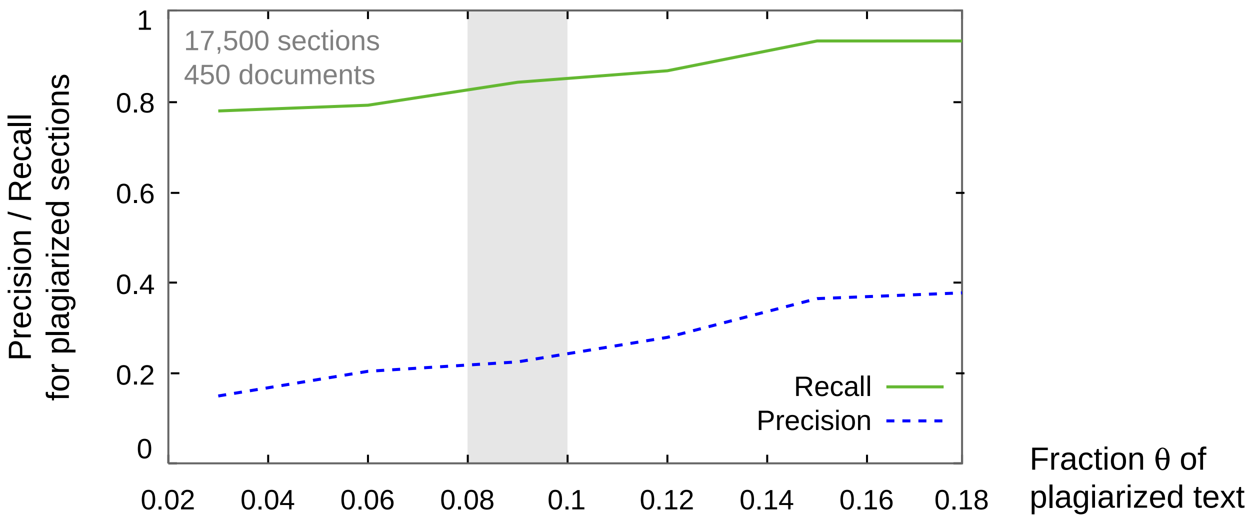


The unsatisfying precision is rooted in the class imbalance.

The Gretchenfrage: Are parts of d plagiarized, if we find an outlier?

# Outliers	Strategy	$\rightarrow$	Hypothesis
0	minimum risk	$\rightarrow$	not plagiarized
1	minimum risk	$\rightarrow$	plagiarized
2	minimum risk	$\rightarrow$	plagiarized
3	minimum risk	$\rightarrow$	plagiarized

Evaluation: Style Model Performance



The unsatisfying precision is rooted in the class imbalance.

The Gretchenfrage: Are parts of d plagiarized, if we find an outlier?

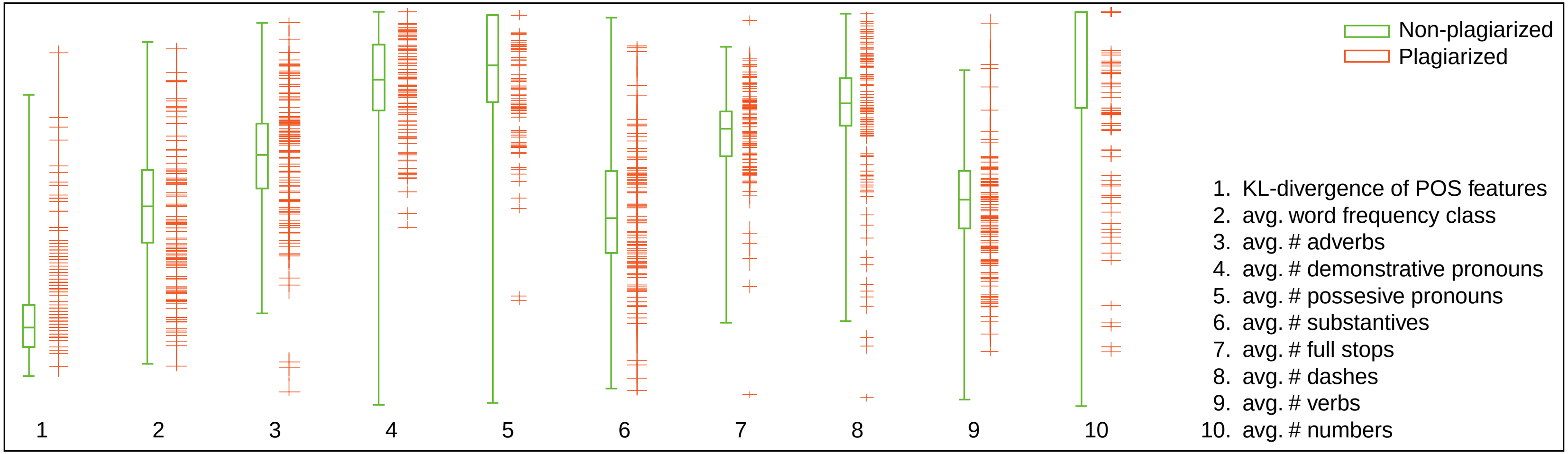
# Outliers	Strategy	→	Hypothesis
0	minimum risk	→	not plagiarized
1	minimum risk	→	plagiarized
2	minimum risk	→	plagiarized
3	minimum risk	→	plagiarized

Strategy	→	Hypothesis
post-processing	→	not plagiarized
post-processing	→	not plagiarized
post-processing	→	not plagiarized
post-processing	→	plagiarized

Intrinsic Analysis and Authorship Verification

Evaluation: Style Model Performance

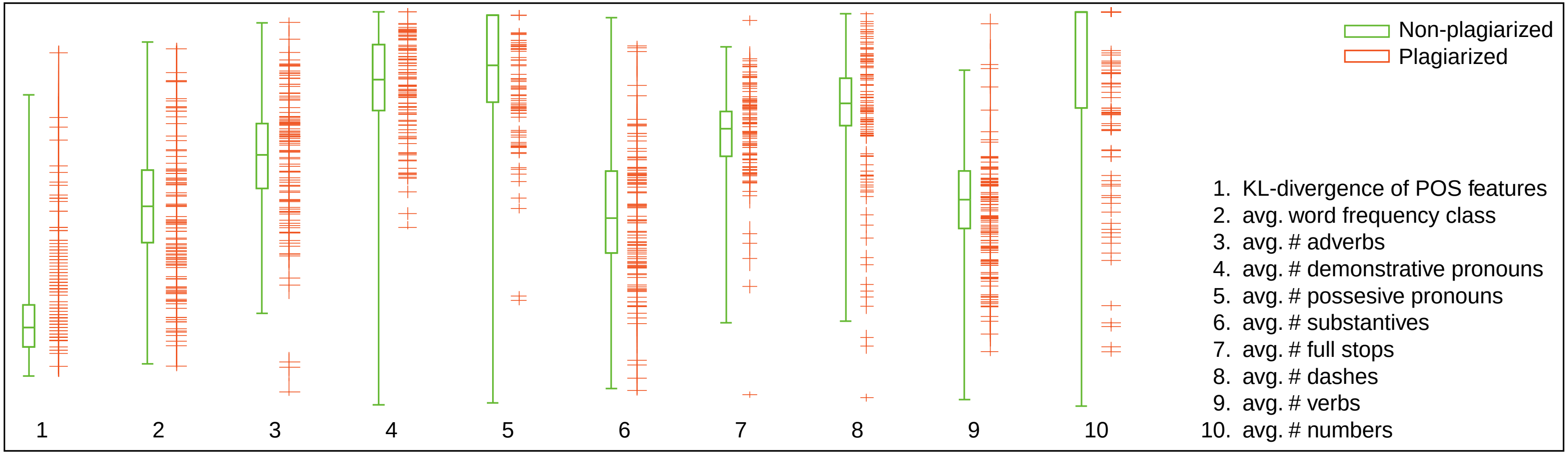
Box plots of 10 style features. 16,000 non-plagiarized target sections and 1,500 outlier sections:



Intrinsic Analysis and Authorship Verification

Evaluation: Style Model Performance

Box plots of 10 style features. 16,000 non-plagiarized target sections and 1,500 outlier sections:



The best performing style features:

Ranking	Feature	Wilk's Lambda	F-Ratio	significant
1	average word frequency class	0.723	152.6	yes
2	average preposition number	0.866	61.4	yes
3	average sentence length	0.880	54.0	yes

Intrinsic Analysis and Authorship Verification

Evaluation: Reliability, Stability

Most stylometric features are designed for analyses at the document level.

Required are those features that are stable at the paragraph level, in order to identify style variations within short texts (6-12 pages).

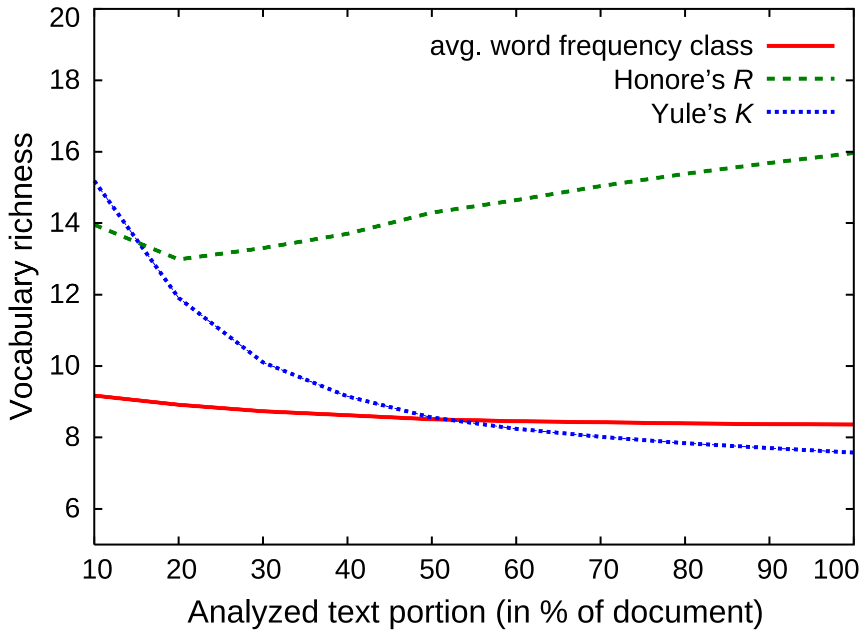
Stylometric feature	Unit of measure	"Unit of reliability"
average paragraph length	paragraph	document
Flesch index	document	document
average sentence length	sentence	paragraph
average word length	word	paragraph
average word frequency class	word	paragraph

Intrinsic Analysis and Authorship Verification

Evaluation: Reliability, Stability

Most stylometric features are designed for analyses at the document level.

Required are those features that are stable at the paragraph level, in order to identify style variations within short texts (6-12 pages).



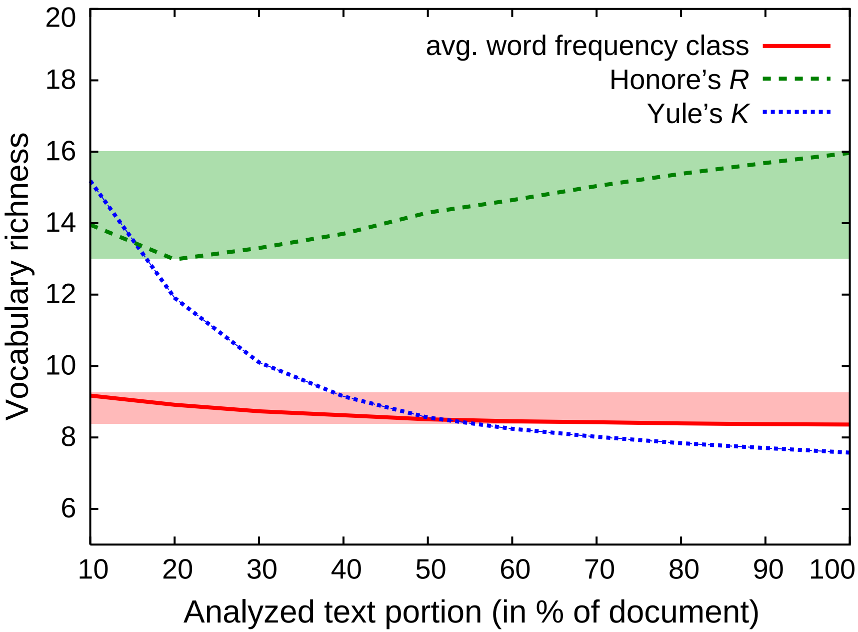
[ECIR, GFKL 2006]

Intrinsic Analysis and Authorship Verification

Evaluation: Reliability, Stability

Most stylometric features are designed for analyses at the document level.

Required are those features that are stable at the paragraph level, in order to identify style variations within short texts (6-12 pages).



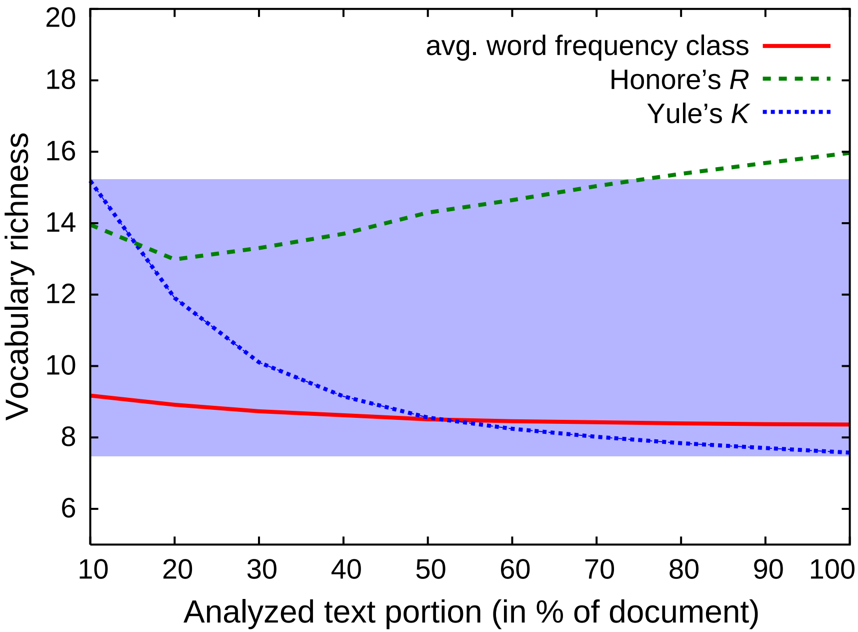
[ECIR, GFKL 2006]

Intrinsic Analysis and Authorship Verification

Evaluation: Reliability, Stability

Most stylometric features are designed for analyses at the document level.

Required are those features that are stable at the paragraph level, in order to identify style variations within short texts (6-12 pages).



[ECIR, GFKL 2006]

Post-Processing with Unmasking [Building Blocks]

Post-Processing with Unmasking

Reliable Interpretation of Outliers

Problem. AVOUTLIER (an easier variant of AVFIND)

Given. A set of texts $D = \{d_1, \ldots, d_n\}$, allegedly written by author A.

Question. Does D contain texts written by an author B, $B \neq A$?

Post-Processing with Unmasking

Reliable Interpretation of Outliers

Problem. AVOUTLIER (an easier variant of AVFIND)
Given. A set of texts $D = \{d_1, \ldots, d_n\}$, allegedly written by author A.
Question. Does D contain texts written by an author B, $B \neq A$?

The belief into an answer depends on the number of found outliers:

# Outliers	Strategy	$\rightarrow$	Hypothesis
0	minimum risk, post-processing	$\rightarrow$	not plagiarized
2	minimum risk	$\rightarrow$	plagiarized
2	post-processing	$\rightarrow$	not plagiarized
4	minimum risk, post-processing	$\rightarrow$	plagiarized

Post-Processing with Unmasking

Reliable Interpretation of Outliers

Problem. AVOUTLIER (an easier variant of AVFIND)

Given. A set of texts $D = \{d_1, \ldots, d_n\}$, allegedly written by author A.

Question. Does D contain texts written by an author B, $B \neq A$?

The belief into an answer depends on the number of found outliers:

# Outliers	**Strategy**	$\rightarrow$	**Hypothesis**
0	minimum risk, post-processing	$\rightarrow$	not plagiarized
2	minimum risk	$\rightarrow$	plagiarized
2	post-processing	$\rightarrow$	not plagiarized
4	minimum risk, post-processing	$\rightarrow$	plagiarized

Post-process borderline situations to gain further evidence for accepting or rejecting a hypothesis.

Idea: Interpret AVOUTLIER results under the Unmasking framework.

Post-Processing with Unmasking

Unmasking for Authorship Verification [Koppel/Schler 2004]

Problem. AV

Given. Two documents d_1, d_2.

Question. Are d_1 and d_2 written by the same author?

Procedure Unmasking:

1. *Chunking.*
2. *Model Fitting.*
3. *Impairing.*
4. Goto Step 2 until the feature space is sufficiently reduced.

Post-Processing with Unmasking

Unmasking for Authorship Verification [Koppel/Schler 2004]

Problem. AV

Given. Two documents d_1, d_2.

Question. Are d_1 and d_2 written by the same author?

Procedure Unmasking:

1. *Chunking.* Decompose d_1, d_2 into two sets of sections, D_1, D_2.
2. *Model Fitting.* With the 250 most frequent words in d_1, d_2 build a VSM for each s in D_1, D_2. Learn a classifier that discriminates between D_1, D_2.
3. *Impairing.* Drop the 3 most discriminating features from the VSMs.
4. Goto Step 2 until the feature space is sufficiently reduced.

Post-Processing with Unmasking

Unmasking for Authorship Verification [Koppel/Schler 2004]

Problem. AV
Given. Two documents d_1, d_2.

Question. Are d_1 and d_2 written by the same author?

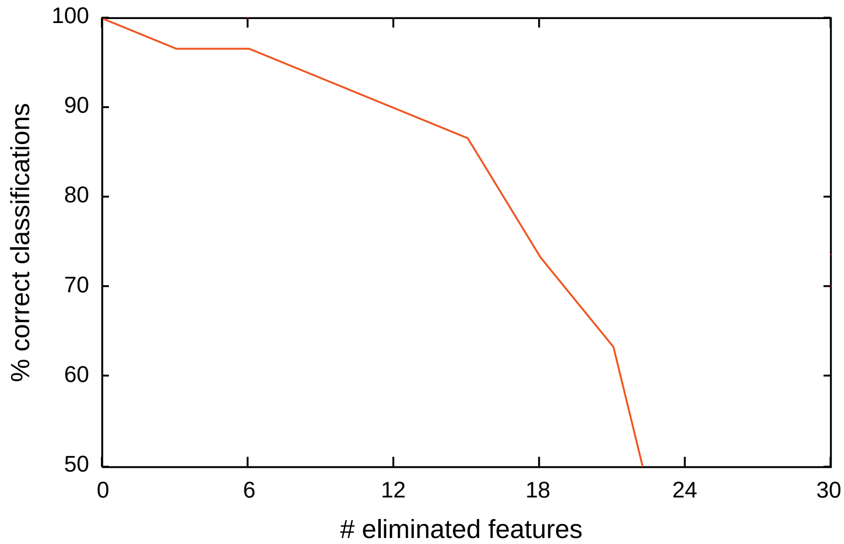
Procedure Unmasking:

1. *Chunking.* Decompose d_1, d_2 into two sets of sections, D_1, D_2.
2. *Model Fitting.* With the 250 most frequent words in d_1, d_2 build a VSM for each s in D_1, D_2. Learn a classifier that discriminates between D_1, D_2.
3. *Impairing.* Drop the 3 most discriminating features from the VSMs.
4. Goto Step 2 until the feature space is sufficiently reduced.
5. *Meta Learning.* Analyze the degradation in the quality of the model fitting.

Post-Processing with Unmasking

Unmasking for Authorship Verification

Characteristic of a typical outcome:



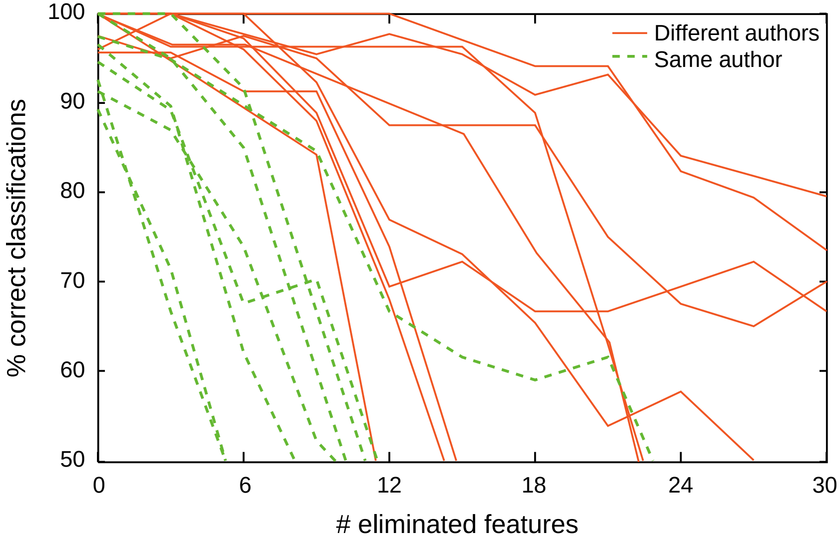
Rationale:

- A large fraction of the 250 words are function words and stop words.
- Only few of the words are related to topic.
- Only few words do the discrimination job—the topic words for a large part.
- Different authors can be distinguished by their use of function words.

Post-Processing with Unmasking

Unmasking for Authorship Verification

Characteristic of a typical outcome:



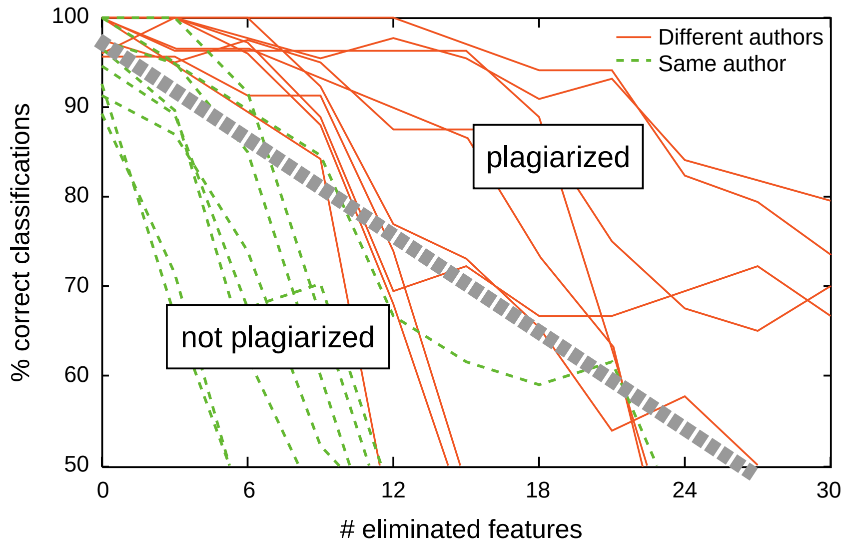
Rationale:

- A large fraction of the 250 words are function words and stop words.
- Only few of the words are related to topic.
- Only few words do the discrimination job—the topic words for a large part.
- Different authors can be distinguished by their use of function words.

Post-Processing with Unmasking

Unmasking for Authorship Verification

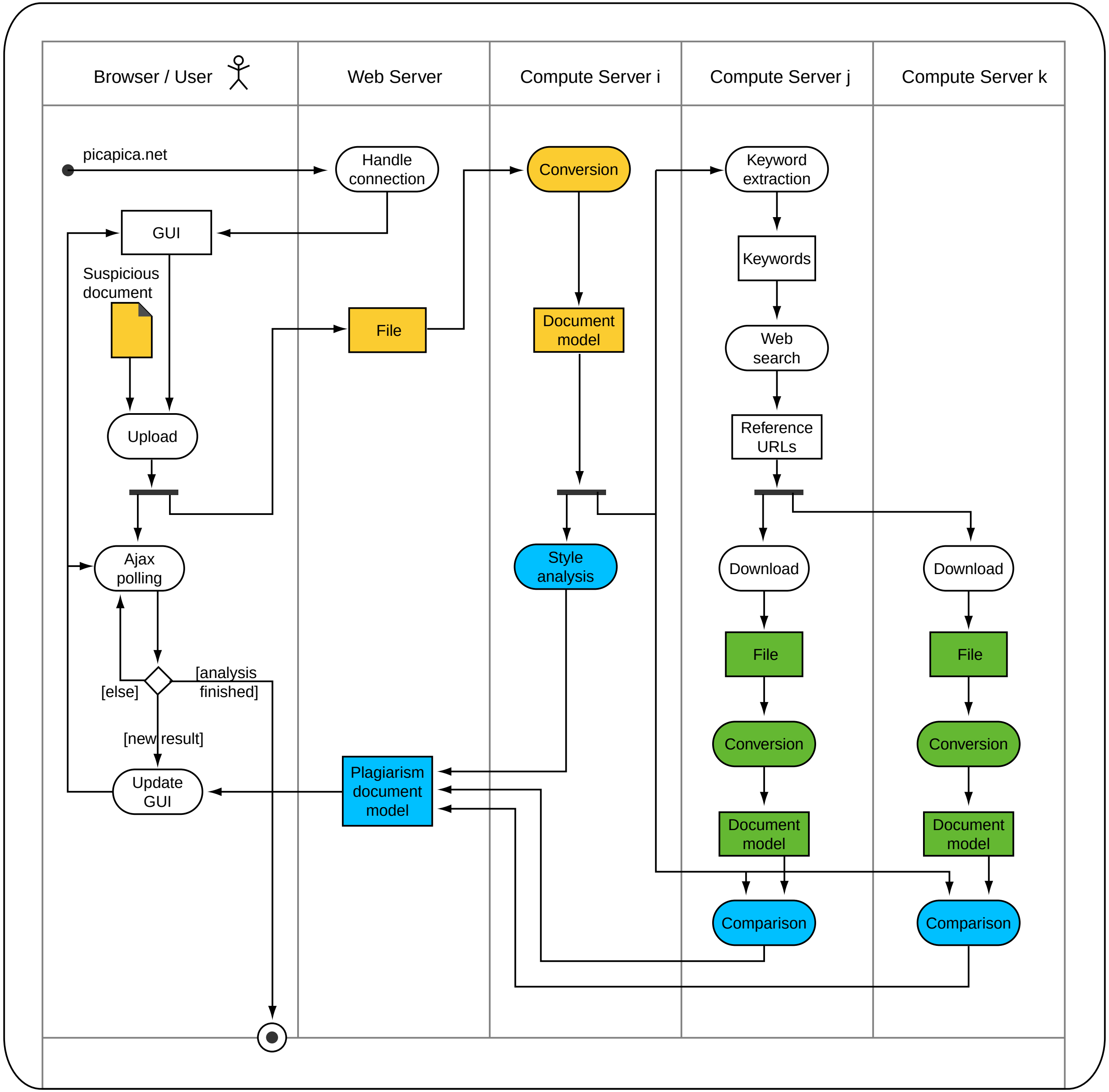
Characteristic of a typical outcome:



Rationale:

- A large fraction of the 250 words are function words and stop words.
- Only few of the words are related to topic.
- Only few words do the discrimination job—the topic words for a large part.
- Different authors can be distinguished by their use of function words.

Software


Browser / User
Web Server
Compute Server i
Compute Server j
Compute Server k
picapica.net
Handle connection
Conversion
Keyword extraction
GUI
Keywords
Suspicious document
File
Document model
Web search
Upload
Reference URLs
Ajax polling
Style analysis
Download
Download
File
File
[analysis finished]
[else]
Conversion
Conversion
[new result]
Update GUI
Plagiarism document model
Document model
Document model
Comparison
Comparison

picapica.net