

On Web-based Plagiarism Analysis

A. Kleppe, D. Braunsdorf, Chr. Lössnitz, S. Meyer zu Eissen

Bauhaus University Weimar
Web-based Information Systems

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

What is Plagiarism?

“Plagiarism refers to the use of another’s ideas, information, language, or writing, when done without proper acknowledgment of the original source”

[Wikipedia]

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

What is Plagiarism?

“Plagiarism refers to the use of another’s ideas, information, language, or writing, when done without proper acknowledgment of the original source”

[Wikipedia]

Plagiarism analysis:

Given. A suspicious document.

Task. Find potentially copied parts,
and provide references to original sources.

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

What is Plagiarism?

“Plagiarism refers to the use of another’s ideas, information, language, or writing, when done without proper acknowledgment of the original source”

[Wikipedia]

Plagiarism analysis:

Given. A suspicious document.

Task. Find potentially copied parts,
and provide references to original sources.

Fact: About 50% of the students admit to plagiarize from Internet documents (study on 18,000 students).

[McCabe 2005]

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Current Research on Plagiarism Analysis

Current research is corpus-oriented.

Given. A suspicious document
and a corpus of original documents.

Task. Find potentially copied parts *in the corpus*,
and provide references to original sources.

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Current Research on Plagiarism Analysis

Current research is corpus-oriented.

Given. A suspicious document
and a corpus of original documents.

Task. Find potentially copied parts *in the corpus*,
and provide references to original sources.

Research questions:

- ❑ How can a corpus of potentially original documents be constructed from the Web?
- ❑ Can plagiarized parts be detected *without* a corpus?

Introduction

Technical
Background

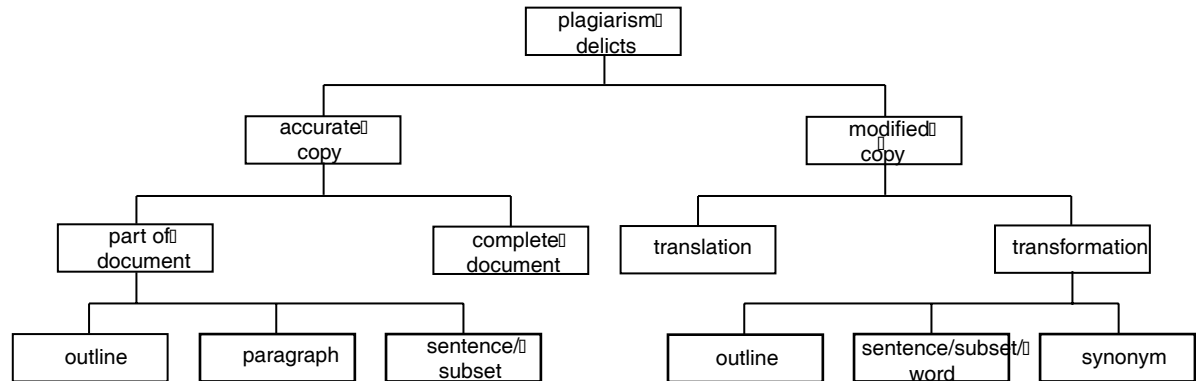
Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Plagiarism Forms

Plagiarism may happen in manifold variants:



Introduction

Technical
Background

Style
Analysis

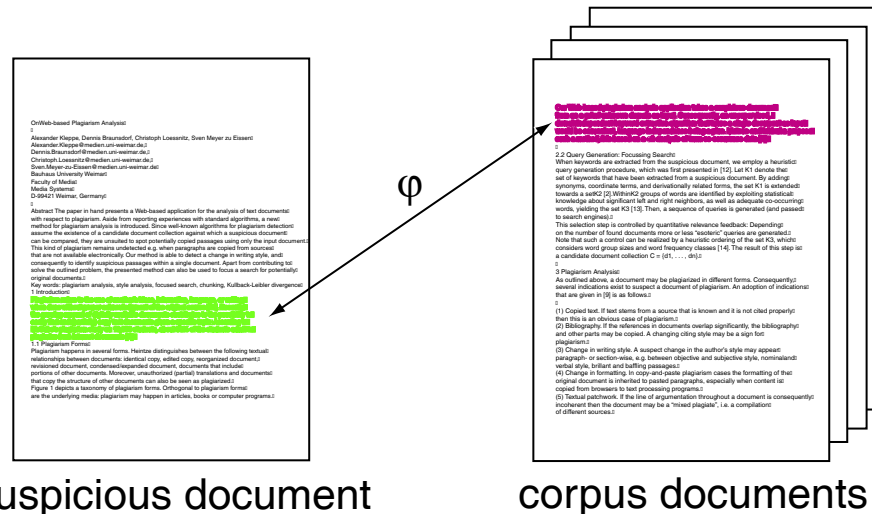
Plagiarism
Detection
on the Web

Prototype

Plagiarism Analysis against a Corpus (1)

Intuition:

- ❑ Partition each document in meaningful units (“chunks”).
- ❑ Compare them with a similarity function φ (pairwise).



Introduction

Technical Background

Style Analysis

Plagiarism Detection on the Web

Prototype

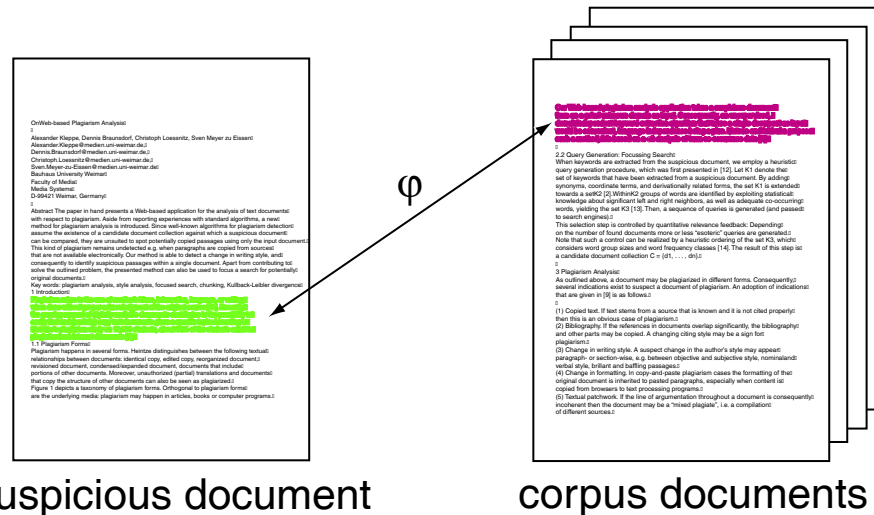
suspicious document

corpus documents

Plagiarism Analysis against a Corpus (1)

Intuition:

- Partition each document in meaningful units (“chunks”).
- Compare them with a similarity function φ (pairwise).



suspicious document

corpus documents

Complexity:

n documents in corpus, c chunks per document on average

$\rightarrow O(n \cdot c^2)$ comparisons

Introduction

Technical Background

Style Analysis

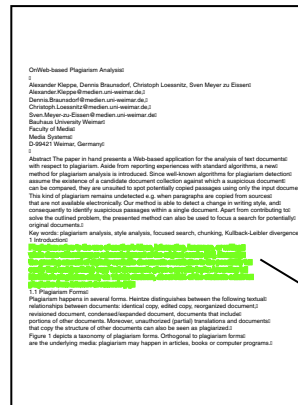
Plagiarism Detection on the Web

Prototype

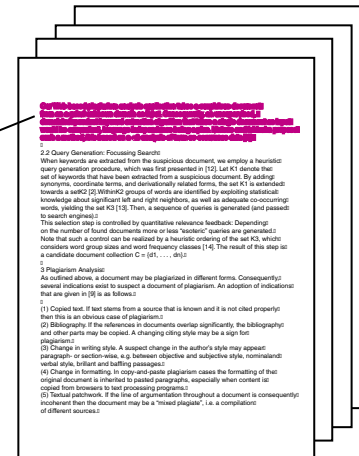
Plagiarism Analysis against a Corpus (2)

Text comparison with digital fingerprints:

- ❑ Partition each document in meaningful units (“chunks”).
- ❑ Compute fingerprints of the chunks using a hash function h .
- ❑ Put all hashes into a hashtable. A collision indicates matching chunks.



suspicious document



corpus documents

Introduction

Technical Background

Style Analysis

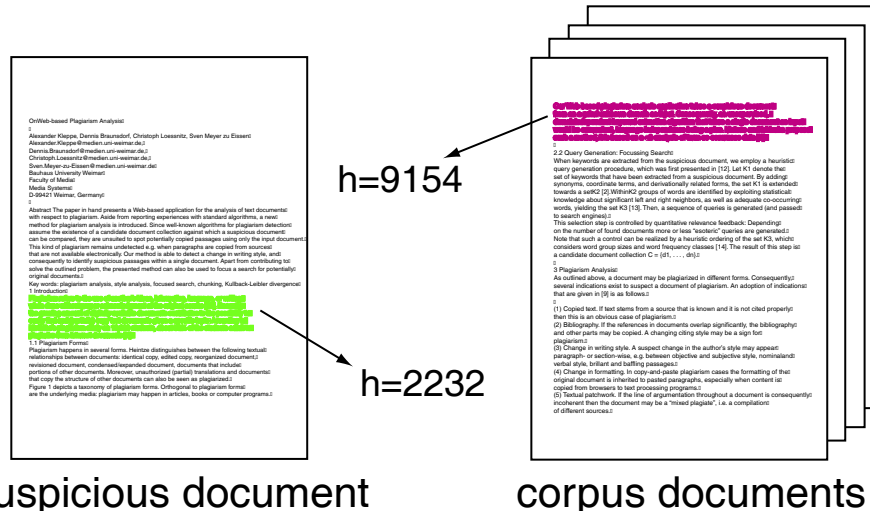
Plagiarism Detection on the Web

Prototype

Plagiarism Analysis against a Corpus (2)

Text comparison with digital fingerprints:

- ❑ Partition each document in meaningful units (“chunks”).
- ❑ Compute fingerprints of the chunks using a hash function h .
- ❑ Put all hashes into a hashtable. A collision indicates matching chunks.



Complexity:

n documents in corpus, c chunks per document on average

→ $O(n \cdot c)$ operations (fingerprint generation, hashtable operations).

Introduction

Technical Background

Style Analysis

Plagiarism Detection on the Web

Prototype

Plagiarism Analysis against a Corpus (3)

Discussion :

- ❑ Hashing is fast, but sensitive to (even small) changes:
 $h(c_1) = h(c_2) \Rightarrow c_1 = c_2$ (with very high probability).
- ❑ Pairwise comparisons based on similarity-function φ are too expensive.
- Past research focussed on chunking strategies.

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Plagiarism Analysis against a Corpus (3)

Discussion :

- ❑ Hashing is fast, but sensitive to (even small) changes:
 $h(c_1) = h(c_2) \Rightarrow c_1 = c_2$ (with very high probability).
- ❑ Pairwise comparisons based on similarity-function φ are too expensive.
- ➔ Past research focussed on chunking strategies.

Current research:

- ❑ Focus on *fuzzy* hash functions h_F :
 $h_F(c_1) = h_F(c_2) \Rightarrow \varphi(c_1, c_2) \geq 1 - \varepsilon$ [Stein 2005]
- ❑ Fuzzy hash functions allow for big chunk sizes (speed-up) and are not sensitive to changes.

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Indications for Plagiarism

Text similarity is not the only indication for plagiarism.

Indications include:

- ❑ Changes in formatting.
- ❑ *Changes in writing style.*
- ❑ Broken argumentation.
- ❑ Inconsistent spelling.
- ❑ Outmoded diction.

These indications can be detected (by humans) without corpora.

→ How can we operationalize the detection of these indications?

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

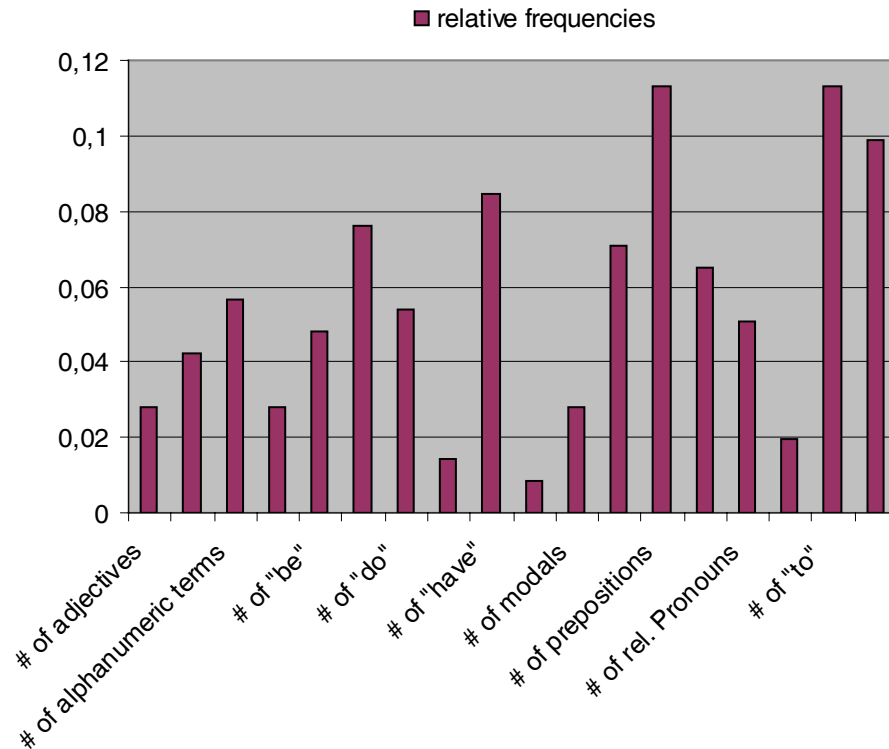
Prototype

Style Analysis

Q: How can writing style changes be measured?

A: Not directly, but divergences of word class distributions give hints.

Word class use in a document:



Introduction

Technical
Background

Style
Analysis

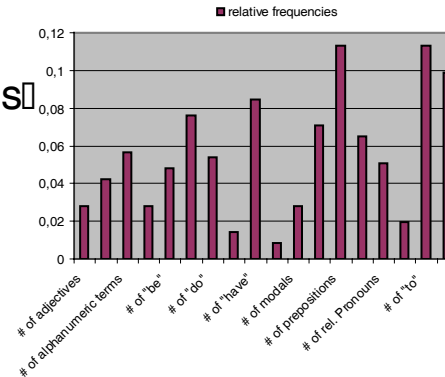
Plagiarism
Detection
on the Web

Prototype

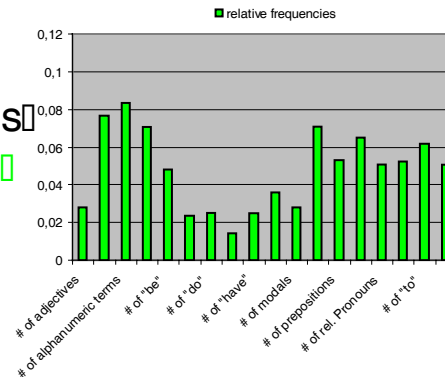
Style Analysis

Idea: Compare the word class distribution of each paragraph to the distribution of the entire document.

word-class frequencies
for the **document**
(global distribution)



word-class frequencies
for a single **paragraph**
(local distribution)



Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Style Analysis

The divergence can be measured by means of the Kullback-Leibler divergence.

Let W denote the set of word classes, let $w \in W$,
let $p(w)$ denote local word class frequencies,
let $q(w)$ denote the global word class frequencies.

Introduction

Technical
Background

**Style
Analysis**

Plagiarism
Detection
on the Web

Prototype

Style Analysis

The divergence can be measured by means of the Kullback-Leibler divergence.

Let W denote the set of word classes, let $w \in W$,
let $p(w)$ denote local word class frequencies,
let $q(w)$ denote the global word class frequencies.

The Kullback-Leibler divergence measure is defined as

$$KL_W(p, q) = \sum_{w \in W} p(w) \log \frac{p(w)}{q(w)} = H(p, q) - H(p) \in \mathbf{R}_0^+$$

If $KL_W(p, q)$ is significant then the paragraph that is associated with p may be copied.

We found KL_W to work very well when single paragraphs are copied from one document to another.

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Web-based Plagiarism Analysis

Given. A suspicious document,
and the Web as corpus of original documents.

Task. Generate a *candidate document base* from the Web,
find potentially copied parts in the base documents,
and provide references to original sources.

Introduction

Technical
Background

Style
Analysis

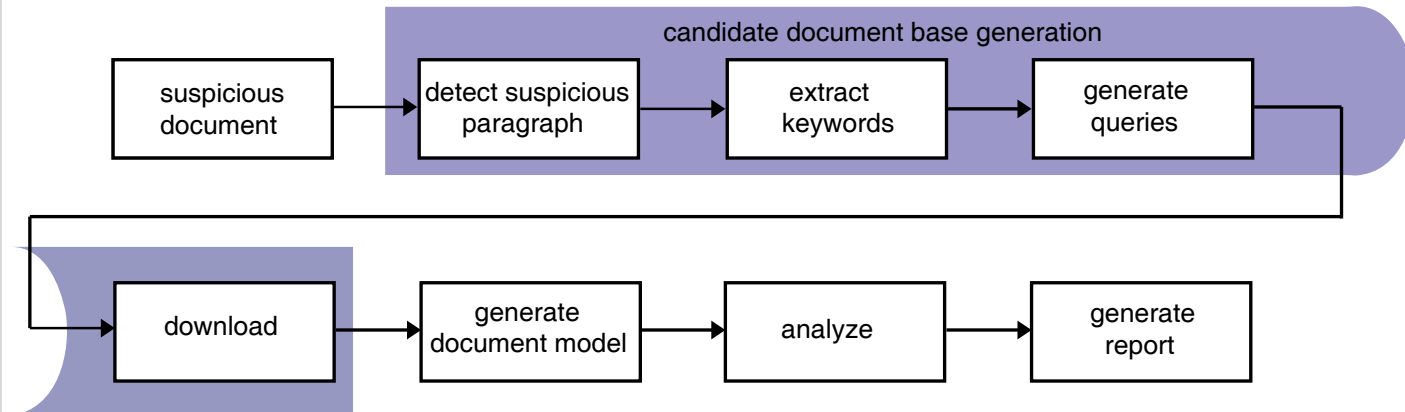
Plagiarism
Detection
on the Web

Prototype

Web-based Plagiarism Analysis

Given. A suspicious document,
and the Web as corpus of original documents.

Task. Generate a *candidate document base* from the Web,
find potentially copied parts in the base documents,
and provide references to original sources.



Introduction

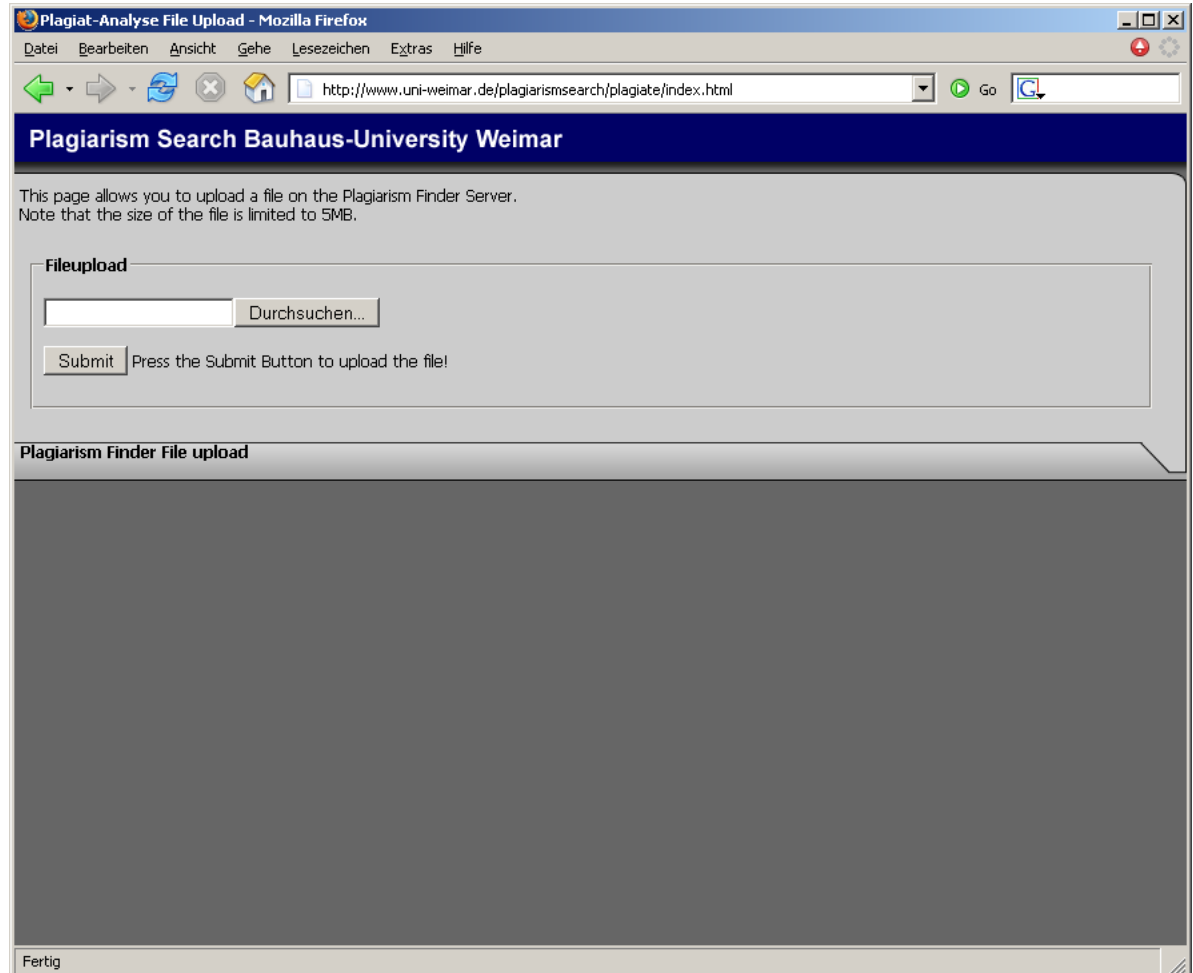
Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Prototypic Implementation



Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Prototypic Implementation

Plagiarismreport - Mozilla Firefox

Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://www.uni-weimar.de/plagiarismsearch/plagate/report/report0.html

Plagiarism Search Bauhaus-University Weimar

Plagiarized Sources 10% 10% 8% 7% 7% 7% 4% 3% 3%

title : inp2.txt	url : opensource.ucc.ie/capi.pdf
words : 3726	title : candidate31.txt
stopwords: 1806	words : 3245
similarity : 0.104806451	stopwords: 1550

Input Document

Source Document

Suspicious paragraphs in document	Divergence
1	0.15
2	0.35
3	0.0
4	0.25
5	0.05

Suspicious paragraph #1 : Divergence = 0.37
Release Management Within Open Source Projects
Abstract
A simple classification system for release management practices is presented. When applied to a set of projects, in this case a set of open source projects, distinctive practices are highlighted and relative strengths can be assessed. Three projects are studied, the Linux kernel, Subversion, and the Apache HTTP server. **Their release practices, as portrayed by the classification system, emerge as a complex combination of subprocesses and tools chosen to support specific project goals and properties.** Through application of this classification.

Evidences in the evolution of OS projects through Changelog Analyses

ABSTRACT
Most empirical studies about Open Source (OS) projects or products are vertical and usually deal with the flagship, successful projects. There is a substantial lack of horizontal studies to shed light on the whole population of projects, including failures. This paper presents a horizontal study aimed at characterizing OS projects.
We analyze a sample of around 400 projects from a popular OS project repository.
Their release practices, as portrayed by the classification system, emerge as a complex combination of subprocesses and tools chosen to support specific project goals and properties. Each project is characterized by a number of attributes. We analyze these attributes statically and over time. The main results show that few projects are capable of attracting a meaningful community of developers. The majority of projects is made by few (in many cases one) person with a very slow pace of evolution. We then try to observe how many projects count on a substantial number of developers, and analyze those projects more deeply. The goal is to achieve a better insight in the dynamics of open source development.

Fertig

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype

Thank You!

Questions?

Introduction

Technical
Background

Style
Analysis

Plagiarism
Detection
on the Web

Prototype