Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Association Rules

Map/Reduce-Algorithms on Hadoop, SomSe 09

Jossekin Beilharz, Cindy Fähnrich

Übersicht

- Aufgabenstellung
  - Begriffsklärung
- Algorithmus
  - Phrasen effizient finden
  - Anzahl gemeinsamer Fenster berechnen
  - Support, Confidence berechnen
- Evaluation

Aufgabenstellung

- Gegeben:

  „mental retardation autosomal recessive clinical features uyguner et al mental retardation large clinical uyguner“

  Fenstergröße: 8

  Phrasen:
  mental retardation autosomal
  mental retardation
  clinical features uyguner
  clinical uyguner

- Lokalisieren
- Assoziationsregeln aufstellen

  mental retardation autosomal ←→ clinical features uyguner
  clinical features uyguner ←→ mental retardation
  mental retardation←→ clinical uyguner

Aufgabenstellung

- Support und Confidence der erstellten Assoziationsregeln berechnen

  mental retardation autosomal → clinical features uyguner

  Support: 14,3% , Confidence: 100%

  clinical features uyguner → mental retardation autosomal

  Support: 14,3%, Confidence: 20%

  ….

  clinical features uyguner → mental retardation

  Support: 28,6% , Confidence: 40%

Übersicht

- Aufgabenstellung
  - Begriffsklärung
- Algorithmus
  - Phrasen effizient finden
  - Anzahl gemeinsamer Fenster berechnen
  - Support, Confidence berechnen
- Evaluation

Support und Confidence

Was ist Support?

- Verhältnis zwischen dem Vorkommen einer Assoziationsregel A → B und der Gesamtanzahl der Fenster:

Support = Vorkommen A → B / Gesamtfensteranzahl

Was ist Confidence?

- Verhältnis zwischen dem Vorkommen einer Assoziationsregel A → B und des Vorkommens von Phrase A:

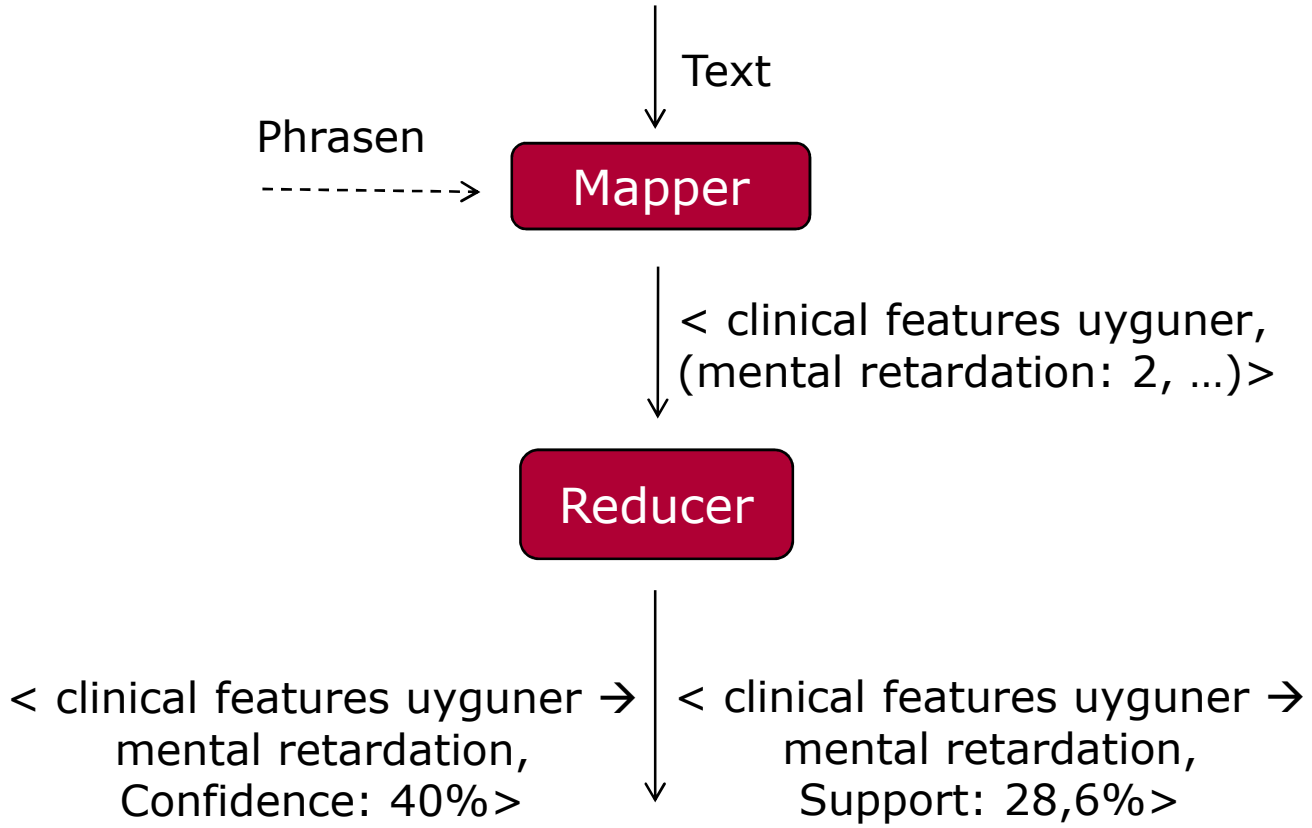
Confidence = Vorkommen A → B / Vorkommen A

Übersicht

- Aufgabenstellung
  - Begriffsklärung
- Algorithmus
  - Phrasen effizient finden
  - Anzahl gemeinsamer Fenster berechnen
  - Support, Confidence berechnen
- Evaluation

Algorithmus

- Abwicklung in einem Map/Reduce-Schritt



Algorithmus

- Mapping
  - Phrasen effizient finden
  - Anzahl gemeinsamer Fenster berechnen
- Reducing
  - „Verschmelzen“ der Key/Value-Pairs
  - Support und Confidence berechnen

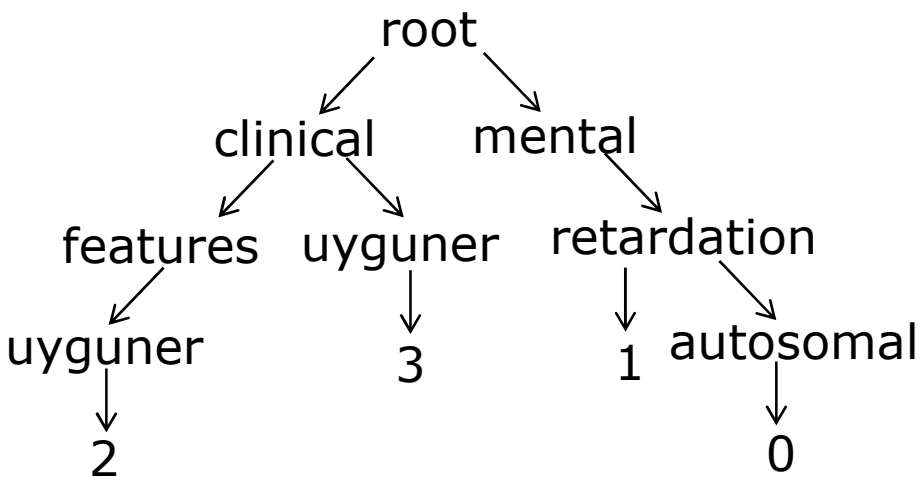
Übersicht

- Aufgabenstellung
  - Begriffsklärung
- Algorithmus
  - Phrasen effizient finden
  - Anzahl gemeinsamer Fenster berechnen
  - Support, Confidence berechnen
- Evaluation

Phrasen effizient finden (Mapper)

- Hohe Ähnlichkeit zwischen Phrasen
- Durchsuchen aller (11 000!) Phrasen bei jedem Wort ineffizient

→ Anordnung der Phrasen in einem Baum:



→ Phrasenindizes als Blätter um Phrasenende zu markieren

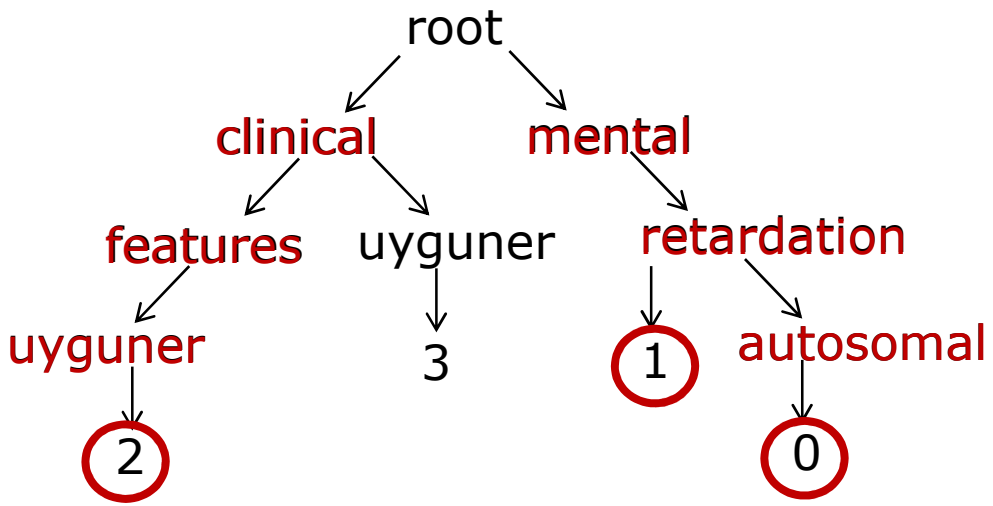
Phrasen effizient finden (Mapper)

- Merker setzen (Wurzel ist immer ein Merker)
- Bei jedem Vergleich mit einem Wort
  - Kindknoten aller Merker auf Gleichheit/Phrasenende prüfen
  - Merker aktualisieren
  - Bei Phrasenende Wert des Blattes und Textdatei-Index merken

    → Speichern in Tupeln:

    [Textdatei-Index, Blattwert]

Phrasen effizient finden (Beispiel)



Tupel:
[1,1]
[2,0]
[6,2]

0 1 2 3 4 5 6 7

„mental retardation autosomal recessive clinical features uyguner et…“

Übersicht

- Aufgabenstellung
  - Begriffsklärung
- Algorithmus
  - Phrasen effizient finden
  - Anzahl gemeinsamer Fenster berechnen
  - Support, Confidence berechnen
- Evaluation

Anzahl gemeinsamer Fenster berechnen (Mapper)

- Für jeden Phrasenfund
  - Fenster um Phrase legen
  - Bereich auf weitere Phrasen untersuchen und Anzahl gemeinsamer Fenster berechnen
  - Speichern in Tupeln:

    [Index der weiteren Phrase, Fensteranzahl]
  - Speichern von zusätzlichen Tupeln mit Indizes -1, -2

  → Notwendig für Support und Confidence
- Key/Value-Pair generieren:

  <Phrasenindex, Tupelmenge>

Anzahl gemeinsamer Fenster berechnen (Beispiel)

„…clinical features uyguner et al mental retardation large …"

mental retardation
Index: 1
Distanz: 6
Gemeinsame Fenster: 8 – 6 = 2
→ Tupel: [1,2]

Zusätzliche Tupel: [-1,5], [-2,7]

Key/Value-Pair: <2, ([1,2],[-1,5],[-2,7])>

Anzahl gemeinsamer Fenster berechnen (Ausnahmen)

- Normale Fensteranzahl mit Phrase verringert sich wenn diese am Anfang/Ende des Textes steht:

„... clinical uyguner ... large clinical uyguner"

Fenster mit Phrase: **6**
(Fenstergröße = 8)

Fenster mit Phrase: **1**
(Fenstergröße = 8)

- Anzahl gemeinsamer Fenster verringert sich bei doppelten Phrasen in einem Fenster:

„... clinical uyguner al mental retardation clinical uyguner ..."

Fenster mit Vorkommen von clinical uyguner, mental retardation: ***7***
(statt 9)

Übersicht

- Aufgabenstellung
  - Begriffsklärung
- Algorithmus
  - Phrasen effizient finden
  - Anzahl gemeinsamer Fenster berechnen
  - Support, Confidence berechnen
- Evaluation

Confidence berechnen (Reducer)

- Key/Value-Pair <2, ([1,2],[-1,5],[-2,7])>
- Confidence

  Confidence
  Vorkommen A → B / Gesamtvorkommen A

  - Gesonderter Eintrag:

    [-1, Anzahl aller Fenster mit Schlüsselphrase]
  - Berechnung:
    - für 2 → 1: 2/5 = 40%
- Output

<clinical features uyguner → mental retardation, Confidence: 40%>

Support berechnen (Reducer)

- Key/Value-Pair: <2, ([1,2],[-1,5],[-2,7])>
- Support

Support:
Vorkommen A → B / Gesamtfensteranzahl

  - Gesonderter Eintrag:

    [-2, Anzahl aller Fenster]
  - Berechnung:
    - für 2 → 1: 2/7 = 28,6%

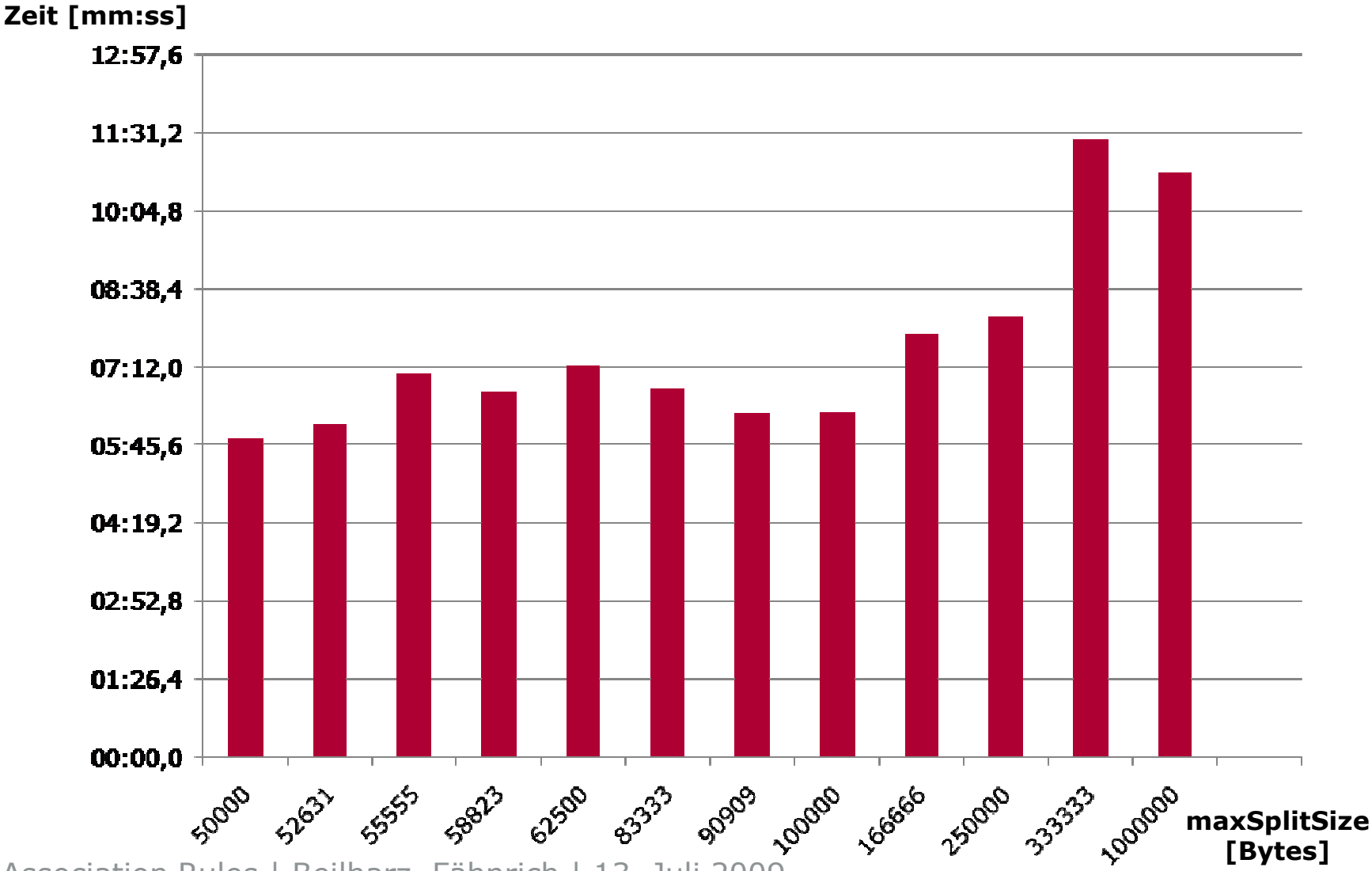
- Output

<clinical features uyguner → mental retardation, Support: 28,6 %>

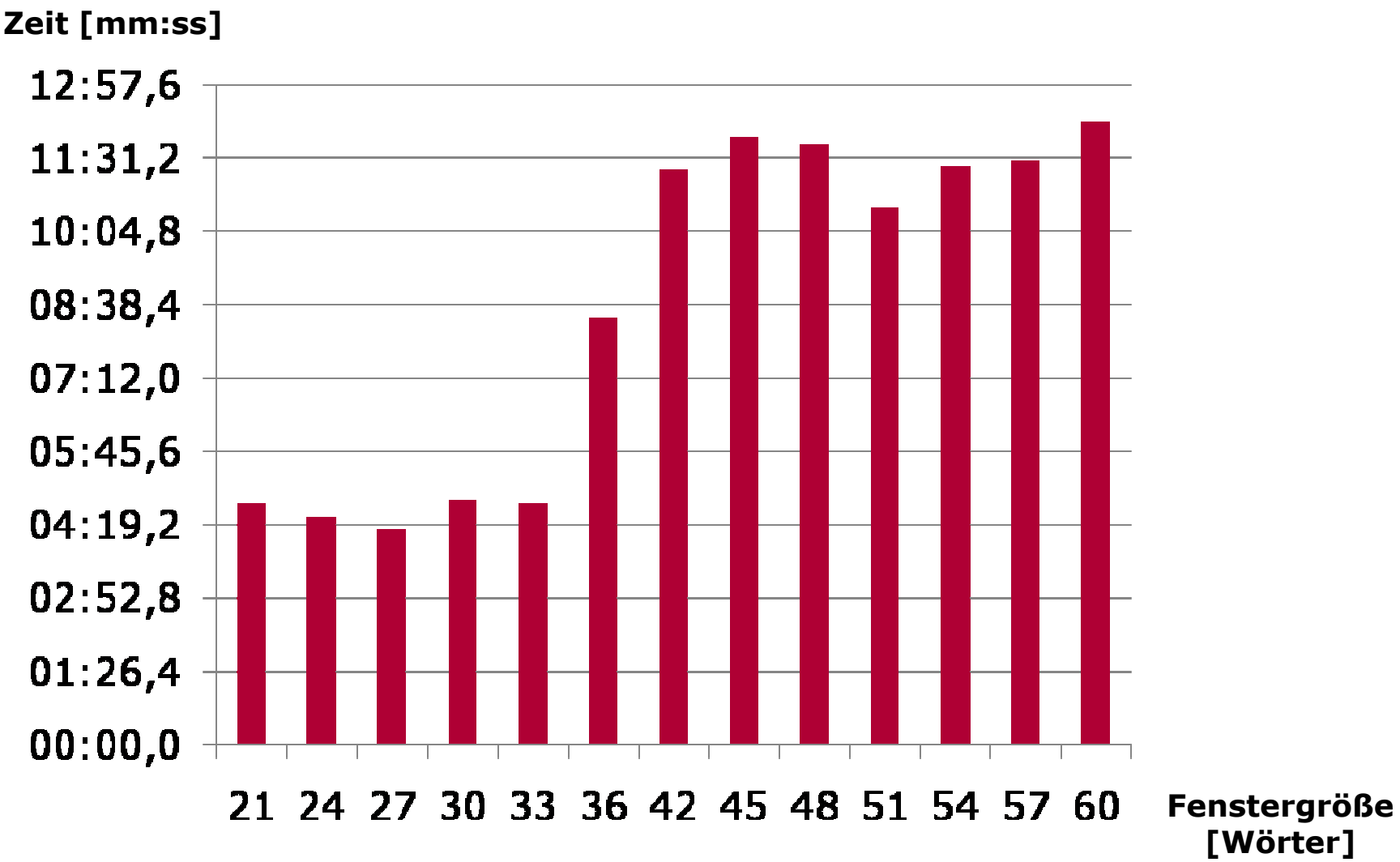
Übersicht

- Aufgabenstellung
  - Begriffsklärung
- Algorithmus
  - Phrasen effizient finden
  - Anzahl gemeinsamer Fenster berechnen
  - Support, Confidence berechnen
- Evaluation

Evaluation


Zeit [mm:ss]
12:57,6
11:31,2
10:04,8
08:38,4
07:12,0
05:45,6
04:19,2
02:52,8
01:26,4
00:00,0
50000
52631
55555
58823
62500
83333
90909
100000
166666
250000
333333
1000000
maxSplitSize [Bytes]

Evaluation


Zeit [mm:ss]
12:57,6
11:31,2
10:04,8
08:38,4
07:12,0
05:45,6
04:19,2
02:52,8
01:26,4
00:00,0
21 24 27 30 33 36 42 45 48 51 54 57 60
Fenstergröße [Wörter]