()BayesLSH-Lite Algorithmus

Input: Eine Menge von Dokumentpaaren C, Ein Ähnlichkeitsgrenzwert t, ein Abrufsparameter a,

Anzahl der zu verwendenden Hashes h

Output: Eine Menge O von Paaren (x, y) zusammen mit genauen Ähnlichkeiten Ŝ x, y

Pseudocode:

import class comparable

int Aehnlichkeitsschaetzung (C, t, a, h)

**Input:** Array von Dokumentenpaare C, Dokumentenpaar I

**Output:** Ähnlichkeitsschätzungen zu jedem Paar (x, y) und ein Array O von Dokumentenpaare (x, y)

**Entitäten aussortieren:** Abbau der falsch positiven Entitätenpaare /Dokumentenpaare

H(x): Hash des Dokuments x aus dem Dokumentenpaar (x, y)

H(y): Hash des Dokuments y aus dem Dokumentenpaar (x, y)

Pr: = Wahrscheinlichkeit, dass die Ähnlichkeit ist großer als der Ähnlichkeitsgrenzwert t  
S: = Ähnlichkeit (zufällige Variable bei Pr)  
p: = posterior Verteilung (Bayes Satz)

C: =Array von I Entitätsobjektpaare/Dokumentenpaare (Menge von Dokumentpaare, die vorher in BD4B auf Ähnlichkeit geprüft waren)

I: = Array von zwei Dokumente (Dokumentenpaar)

: genaue Ähnlichkeit zwischen Dokumente x und y

t: = Ähnlichkeitsgrenzwert für die zu vergleichenden Dokumentenpaare (wird vom Benutzer spezifiziert)

a: = Abrufsparameter (Nicht sicher: die Anzahl an gewünschte Elemente)

h: = Anzahl an Hashes, die zu benutzen sind

int n: = Anzahl an Hashes

int m: = Übereinstimmung von Dokumenten

M (m, n): = Ereignis von m Übereinstimmungen aus den n ersten Hashes

Boolean istaussortiert: Variable zum Prüfen ob ein Entitätspaar/Dokumentenpaar aussortiert wurde oder nicht

String x, y = Entitäten/Dokumenten von Array I

**for all** I [x, y] ∈ C **do**

n = 0

m = 0

istaussortiert ==false

**While** n < h **do**

m = m + Summe von n zu n + k von Hi(x). compareto(Hi(y)) (Summe von allen übereinstimmenden Hashes für ein Dokumentenpaar)

n = n + k

**if** Pr[S>=t|M (m, n)] < a **then**

istaussortiert == true

**break** (sortiere die Dokumentenpaare)

**end if**

**end while**

**if** istaussortiert == false **then**

= aehnlichkeit (x, y) (genaue Ähnlichkeit)

**if**  > t **then**

O. add (I (x, y), )

**end if**

**end for**

**Return** O

(n+k).=

**Discrete Random Variables** Pr[]:

1. Compare the next b hashes and compute the early pruning probability P[S ≥ t| M(m, n)].   
2. If P[S ≥ t| M(m, n)] < α, then prune the pair and stop.  
3. If maximum allowable hash comparisons have been reached, compute exact similarity and stop. 4. Go to step 1