All-Pairs with first n sets preprocessing

Daniel Brand
Alexander Gruschina

Similarity Search PS
WS 2016/17

The basic problem

Given 2 data collections (here: same file twice)

Find “similar” sets (here: lines) with respect to some threshold

Sets = Lines, Tokens of set = Words in line

Define “similarity” ?

Jaccard Similarity of 2 Sets

Overlap $O(x, y) = |x \cap y|$

Jaccard $J(x, y) = \frac{|x \cap y|}{|x \cup y|}$

- Basic example:

  `x:` `Daniel broke the code`

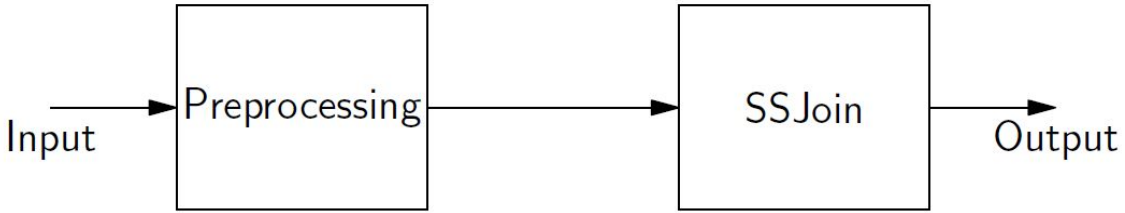
  `y:` `Alex broke the code`

  O(`x`,`y`) = |{`broke`,`the`,`code`}| = 3

  J(`x`,`y`) = O(`x`,`y`) / |{`Daniel`,`broke`,`the`,`code`,`Alex`}| = 3/5 = 0,6

Given Jaccard threshold τ : sets `x`, `y` are *similar* if J(`x`, `y`) ≥ τ

Initial approach



1. Preprocessing on whole data
2. Run fancy algorithm

Problem: Preprocessing very expensive (e.g. sorting)

Algorithms hardly exhibit room for optimizations

Our topic

- Apply non-optimal preprocessing
- Hope for reducing overall runtime

Some more details, please

Before:

1. Scan all sets
2. Determine token frequencies
3. Order sets by length

Some more details, please

Now:

1. Scan only first N sets
2. Determine token frequencies within first N sets
3. Treat tokens in remaining sets as equally “uncommon”
4. No sorting (only in-set sorting on tokens)

Treating tokens uncommon

First n sets:

- If token occurs 4 times in the first n sets → id = 4
- Multiple tokens with same freq → id = freq+1, +2, ...

Remaining sets:

- We don’t calculate frequencies
- Just assign negative ids
- Starting with 0,-1,-2, ...

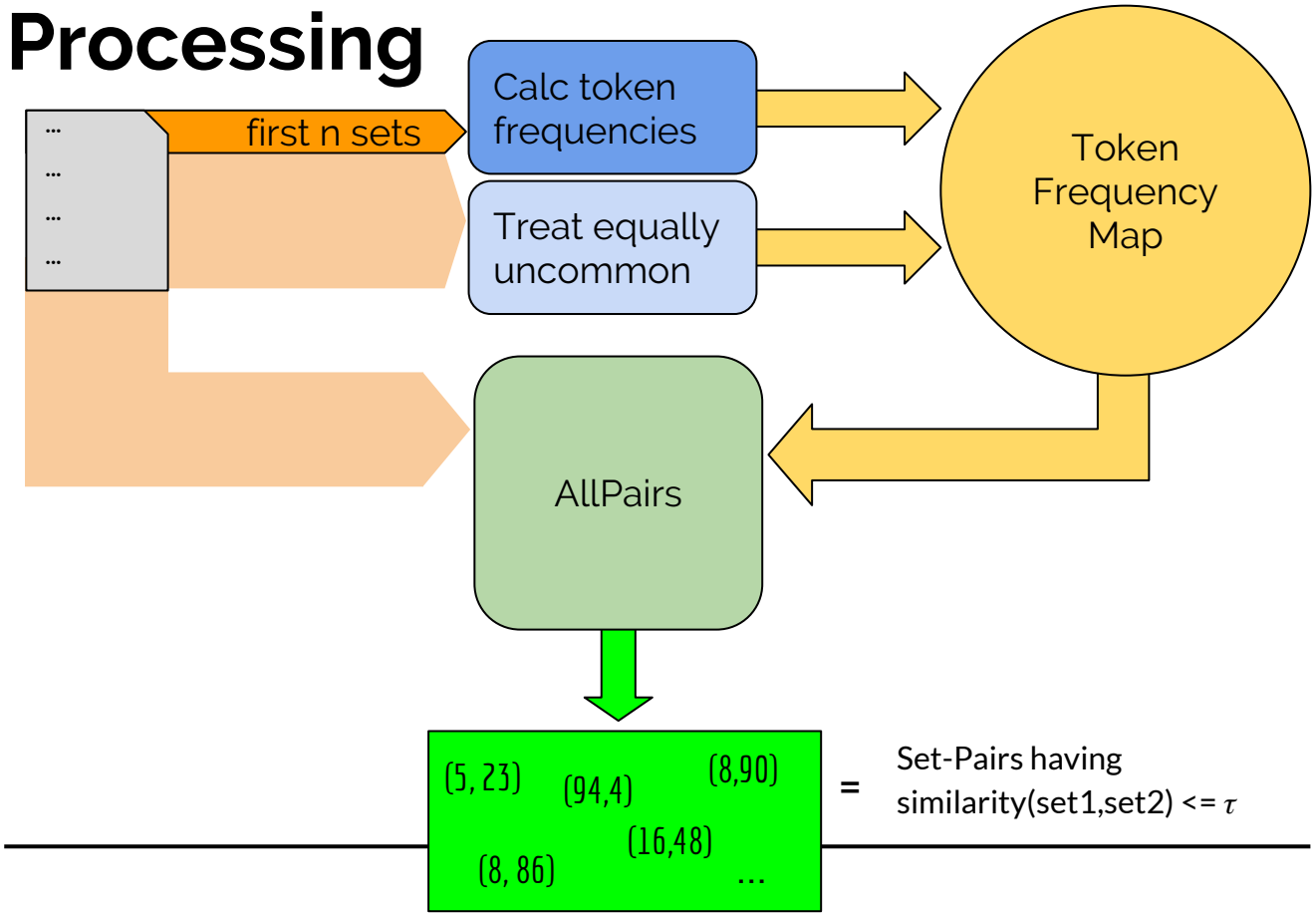
Processing

Parameters

N

filename dblp/enron

Jaccard threshold τ



Performance Analysis

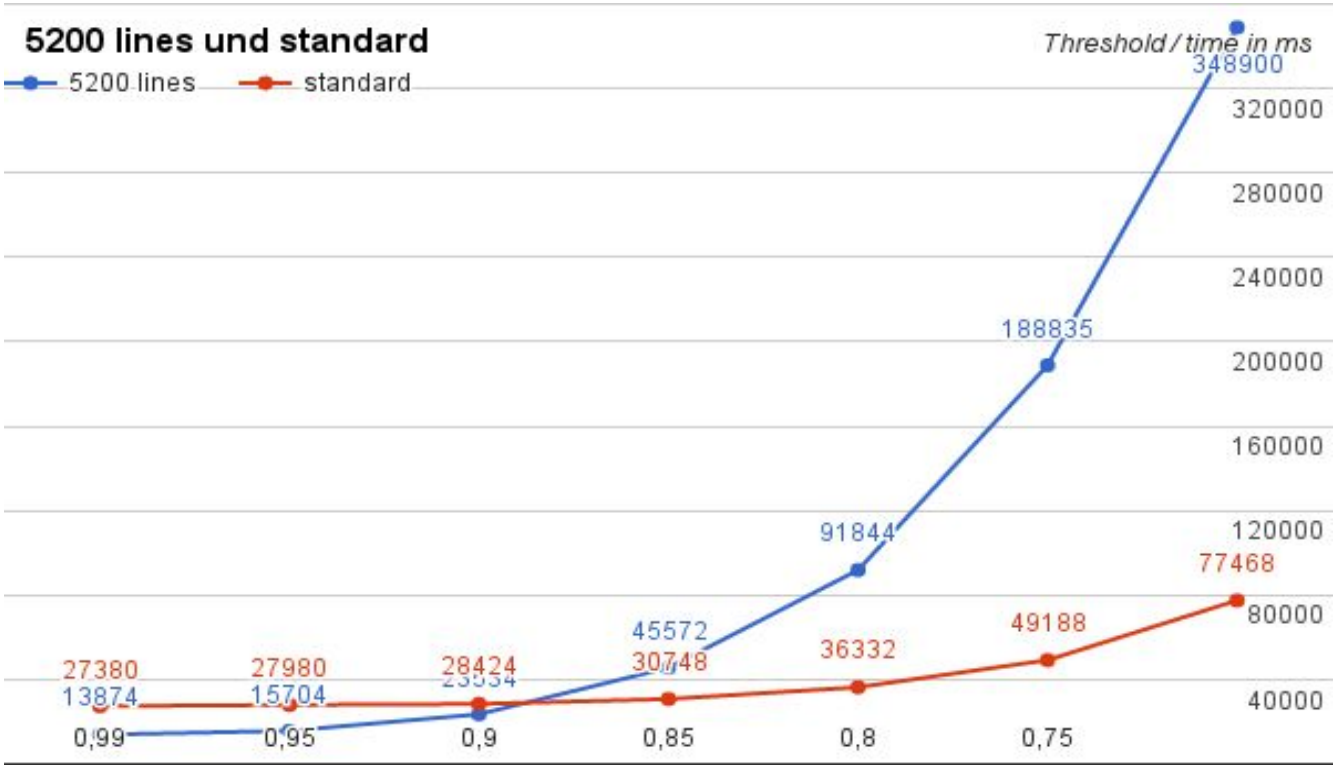
- Vary
  - N
  - Jaccard threshold τ
  - file
- Handle duplicates within sets
  - “Unduplicate” routine applied

    `He broke the code broke → He broke the code broke4`
- Correctness verified
  - Same number of pairs (and same pairs!) found
- Compare runtime with initial approach
- Using *perf* utility

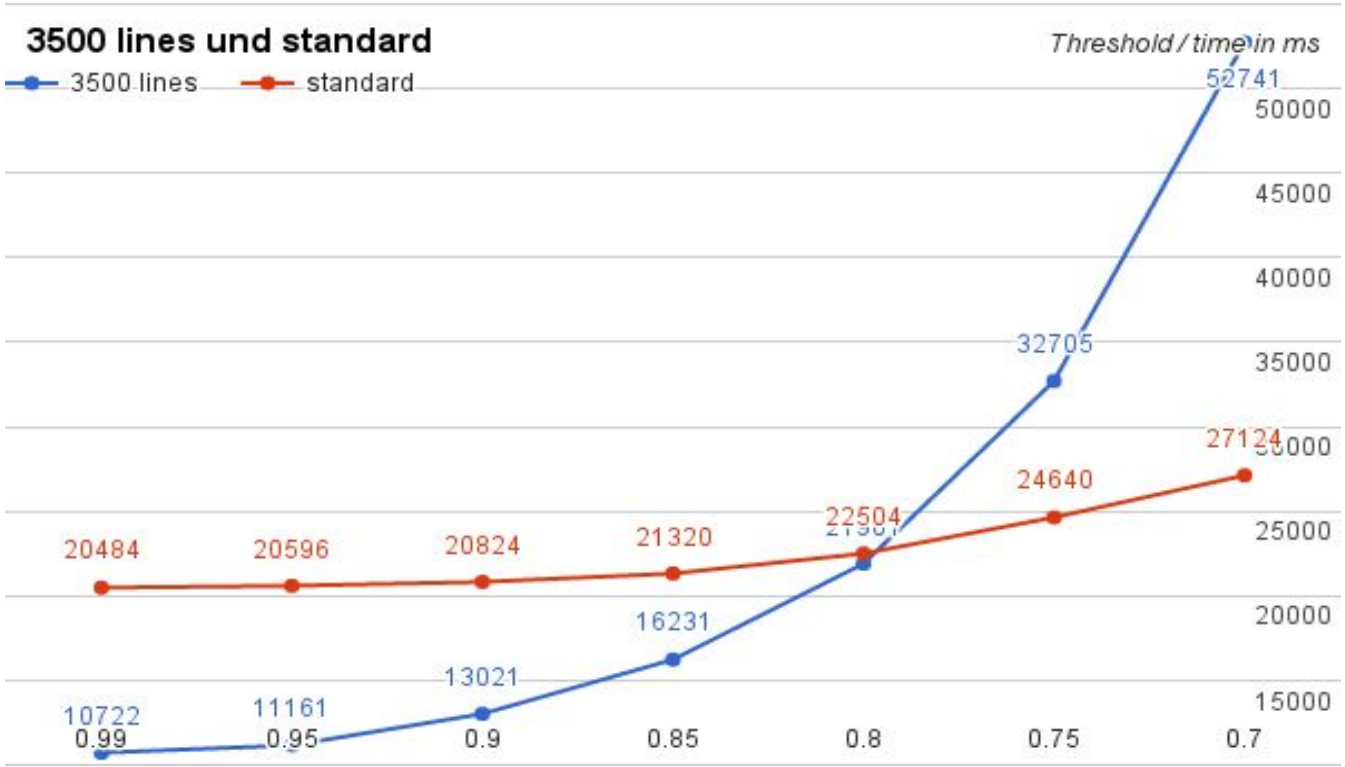
Performance Analysis

- Use ~1% of the lines for prediction
  - enron.format has 517431 lines
    - 1% ≈ 5200 lines
  - trec.txt has 348566 lines
    - 1% ≈ 3500 lines
  - dblp.txt has 3870284 lines
    - 1% ≈ 40000 lines

enron.format



trec.txt without duplicates



dblp.txt without duplicates