Efficient Approximate Search on String Collections
Part I

Marios Hadjieleftheriou

Chen Li

at&t

UCIrvine
University of California, Irvine
Try their names (good luck!)

UCSD
Yannis Papakonstantinou

Case Western
Meral Ozsoyoglu

AT&T--Research
Marios Hadjieleftheriou

http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/index.html
Search Results for 'hadjeleftheriou'

Sorry, no author found.

Copyright © Wed Aug 26 09:31:37 2009 by Michael Ley (ley@uni-trier.de)
Better system?

http://dblp.ics.uci.edu/authors/
People Search at UC Irvine

The goal of the PSearch Project is to make it easier to search for UCI People. It has a single input search box, which allows keyword queries on people name, UCInetID, telephone number, department, and title. More

Search for People & Departments [Advanced Search]

Enter a name, ucinetid, e-mail or phone extension.

Shared MEHROTA smehrotr

Professor [949] 824-5975 Science

Mehran MOSHFEghi mmoshfeg Mehran Adjunct Professor [310] 804-8808 Electrical Engineering & (and) Computer Science

Joerg imeyer Assistant [949] Electrical Engineering & (and) Computer Science
Data Cleaning
Problem Formulation

Find strings similar to a given string: \(\text{dist}(Q,D) \leq \delta\)

Example: find strings similar to “hadjeftheriou”

Performance is important!
- 10 ms: 100 queries per second (QPS)
- 5 ms: 200 QPS
Outline

- Motivation
- Preliminaries
- Trie-based approach
- Gram-based algorithms
- Sketch-based algorithms
- Compression
- Selectivity estimation
- Transformations/Synonyms
- Conclusion

Part I

Part II
Preliminaries
Similarity Functions

- Similar to:
  - a domain-specific function
  - returns a similarity value between two strings

- Examples:
  - Edit distance
  - Hamming distance
  - Jaccard similarity
  - Soundex
  - TF/IDF, BM25, DICE

See [KSS06] for an excellent survey
Edit Distance

- A widely used metric to define string similarity
- \( \text{Ed}(s_1,s_2) = \) minimum # of operations (insertion, deletion, substitution) to change \( s_1 \) to \( s_2 \)
- Example:
  
  \[
  \begin{align*}
  s_1: \text{Tom Hanks} \\
  s_2: \text{Ton Hank}
  \end{align*}
  \]
  
  \( \text{ed}(s_1,s_2) = 2 \)
Next…

Trie-based approach [JLL+09]
Trie Indexing

Strings
exam
example
exemplar
exempt
sample
Active nodes on Trie

Query: “example”
Edit-distance threshold = 2

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>examp</td>
<td>2</td>
</tr>
<tr>
<td>exampl</td>
<td>1</td>
</tr>
<tr>
<td>example</td>
<td>0</td>
</tr>
<tr>
<td>exempl</td>
<td>2</td>
</tr>
<tr>
<td>exempla</td>
<td>2</td>
</tr>
<tr>
<td>sample</td>
<td>2</td>
</tr>
</tbody>
</table>
**Initialization**

- $Q = \varepsilon$

![Diagram showing initial active nodes within depth $\delta$.](image)

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>0</td>
</tr>
<tr>
<td>e</td>
<td>1</td>
</tr>
<tr>
<td>ex</td>
<td>2</td>
</tr>
<tr>
<td>s</td>
<td>1</td>
</tr>
<tr>
<td>sa</td>
<td>2</td>
</tr>
</tbody>
</table>

Initial active nodes: all nodes within depth $\delta$
Incremental Algorithm

Return $\xi_n$ nodes as answers.
Q = e x a m p l e

Active nodes for Q = e

Prefix Distance

- ε 0
- e 1
- ex 2
- s 1
- sa 2

Active nodes for Q = e

Prefix # Op Base Op

<table>
<thead>
<tr>
<th>Prefix</th>
<th># Op</th>
<th>Base</th>
<th>Op</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</td>
<td>1</td>
<td>ε</td>
<td>del e</td>
</tr>
<tr>
<td>s</td>
<td>1</td>
<td>ε</td>
<td>sub e/s</td>
</tr>
<tr>
<td>e</td>
<td>0</td>
<td>ε</td>
<td>mat e</td>
</tr>
<tr>
<td>ex</td>
<td>1</td>
<td>ε</td>
<td>ins x</td>
</tr>
<tr>
<td>exa</td>
<td>2</td>
<td>ε</td>
<td>Ins xa</td>
</tr>
<tr>
<td>exe</td>
<td>2</td>
<td>ε</td>
<td>Ins xe</td>
</tr>
<tr>
<td>sa</td>
<td>3</td>
<td>sa</td>
<td>del e</td>
</tr>
</tbody>
</table>
Good and bad

- Advantages:
  - Trie size is small
  - Can do search as the user types

- Disadvantages
  - Works for edit distance only
Gram-based algorithms

- List-merging algorithms [LLL08]
- Variable-length grams (VGRAM) [LWY07,YWL08]
“q-grams” of strings

universal

2-grams
Edit operation’s effect on grams

$k$ operations could affect $k \times q$ grams

If $ed(s_1, s_2) \leq k$, then their # of common grams $\geq (|s_1| - q + 1) - k \times q$
q-gram inverted lists

<table>
<thead>
<tr>
<th>id</th>
<th>strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>rich</td>
</tr>
<tr>
<td>1</td>
<td>stick</td>
</tr>
<tr>
<td>2</td>
<td>stich</td>
</tr>
<tr>
<td>3</td>
<td>stuck</td>
</tr>
<tr>
<td>4</td>
<td>static</td>
</tr>
</tbody>
</table>

2-grams
Searching using inverted lists

- Query: “shtick”, ED(shtick, ?) ≤ 1

sh  ht  ti  ic  ck

# of common grams >= 3

<table>
<thead>
<tr>
<th>id</th>
<th>strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>rich</td>
</tr>
<tr>
<td>1</td>
<td>stick</td>
</tr>
<tr>
<td>2</td>
<td>stich</td>
</tr>
<tr>
<td>3</td>
<td>stuck</td>
</tr>
<tr>
<td>4</td>
<td>static</td>
</tr>
</tbody>
</table>
T-occurrence Problem

Find elements whose occurrences \( \geq T \)
Example

- $T = 4$

\[\begin{array}{c}
1 \\
3 \\
5 \\
10 \\
13
\end{array} \quad \begin{array}{c}
10 \\
13 \\
15
\end{array} \quad \begin{array}{c}
5 \\
7 \\
13
\end{array} \quad \begin{array}{c}
13 \\
15
\end{array}\]

Result: 13
List-Merging Algorithms

HeapMerger → MergeOpt

ScanCount → MergeSkip → DivideSkip

[SK04]
[LLL08, BK02]
Heap-based Algorithm

Count # of occurrences of each element using a heap
MergeOpt Algorithm [SK04]

Binary search

Long Lists: T-1

Short Lists
Example of MergeOpt

<table>
<thead>
<tr>
<th>1</th>
<th>10</th>
<th>5</th>
<th>13</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>13</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Long Lists: 3
Short Lists: 2

Count threshold $T \geq 4$
ScanCount

String ids

| 1  | 0 |
| 2  | 0 |
| 3  | 0 |
| ...| ...|
| 13 | 0 |
| 14 | 0 |
| 15 | 0 |

Increment by

1 1 10 5 13 15
3 13 7
5 15 13
10

Result! 13

# of occurrences

Count threshold $T \geq 4$
List-Merging Algorithms

HeapMerger \rightarrow \text{MergeOpt}

ScanCount \rightarrow \text{MergeSkip} \rightarrow \text{DivideSkip}

[SK04] [LLL08, BK02]
MergeSkip algorithm \cite{BK02, LLL08}
Example of MergeSkip

\[
\text{minHeap} \\
1 \\
5 \quad 10 \\
13 \quad 15 \\
\rightarrow 1 \rightarrow 10 \rightarrow 5 \rightarrow 13 \rightarrow 15 \\
3 \quad 13 \quad 7 \\
5 \quad 15 \quad 17 \\
10 \quad 15 \\
\text{Jump}
\]

Count threshold \( T \geq 4 \)
DivideSkip Algorithm [LLL08]
How many lists are treated as long lists?
Length Filtering

s:

Length: 10

By length only!

Ed(s,t) ≤ 2

Length: 19

t:
Positional Filtering

Ed(s,t) ≤ 2

s: a b

(t, 1)

a b

(t, 12)

(ab, 1)

(ab, 12)
A filter tree

Combine filters with list-merging algorithms [LLL08]
Variable-length grams (VGRAM) [LWY07,YWL08]
2-grams -> 3-grams?

- Query: “shtick”, ED(shtick, ?) ≤ 1

sht  hti  tic  ick

# of common grams ≥ 1

<table>
<thead>
<tr>
<th>id</th>
<th>strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>rich</td>
</tr>
<tr>
<td>1</td>
<td>stick</td>
</tr>
<tr>
<td>2</td>
<td>stich</td>
</tr>
<tr>
<td>3</td>
<td>stuck</td>
</tr>
<tr>
<td>4</td>
<td>static</td>
</tr>
</tbody>
</table>

3-grams:

- ati → 4
- ich → 0 → 2
- ick → 1
- ric → 0
- sta → 4
- sti → 1 → 2
- stu → 3
- tat → 4
- tic → 1 → 2 → 4
- tuc → 3
- uck → 3
Observation 1: dilemma of choosing “q”

- Increasing “q” causing:
  - Longer grams → Shorter lists
  - Smaller # of common grams of similar strings

<table>
<thead>
<tr>
<th>id</th>
<th>strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>rich</td>
</tr>
<tr>
<td>1</td>
<td>stick</td>
</tr>
<tr>
<td>2</td>
<td>stich</td>
</tr>
<tr>
<td>3</td>
<td>stuck</td>
</tr>
<tr>
<td>4</td>
<td>static</td>
</tr>
</tbody>
</table>
Observation 2: skew distributions of gram frequencies

- DBLP: 276,699 article titles
- Popular 5-grams: ation (>114K times), tions, ystem, catio
VGRAM: Main idea

- Grams with **variable lengths** (between $q_{\text{min}}$ and $q_{\text{max}}$)
  - zebra
    - ze(123)
  - corrasion
    - co(5213), cor(859), corr(171)

- Advantages
  - Reduce index size 😊
  - Reducing running time 😊
  - Adoptable by many algorithms 😊
Challenges

- Generating variable-length grams?
- Constructing a high-quality gram dictionary?
- Relationship between string similarity and their gram-set similarity?
- Adopting VGRAM in existing algorithms?
Challenge 1: String $\rightarrow$ Variable-length grams?

- Fixed-length 2-grams

  - Universal [2,4]-gram dictionary

- Variable-length grams
Representing gram dictionary as a trie
Step 2: Constructing a gram dictionary

- Frequency-based [LYW07]
- Cost-based [YLW08]
Challenge 3: Edit operation’s effect on grams

$k$ operations could affect $k \times q$ grams

Fixed length: $q$
Deletion affects variable-length grams

Deletion

\[ i - q_{\text{max}} + 1 \quad \text{Not affected} \]

\[ i \quad \text{Affected} \]

\[ i + q_{\text{max}} - 1 \quad \text{Not affected} \]

\[ \text{Deletion} \]
Main idea

- For a string, for each position, compute the number of grams that could be destroyed by an operation at this position.
- Compute number of grams possibly destroyed by \( k \) operations.
- Store these numbers (for all data strings) as part of the index.

Vector of \( s = <2,4,6,8,9> \)

With 2 edit operations, at most 4 grams can be affected.

- Use this number to do count filtering.
Summary of VGRAM index

<table>
<thead>
<tr>
<th>id</th>
<th>string</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>stick</td>
</tr>
<tr>
<td>1</td>
<td>stich</td>
</tr>
<tr>
<td>2</td>
<td>such</td>
</tr>
<tr>
<td>3</td>
<td>stuck</td>
</tr>
</tbody>
</table>

(a) strings

(b) Gram dictionary as a trie

(c) Reversed-gram trie

(d) NAG vectors

<table>
<thead>
<tr>
<th>id</th>
<th>NAG vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2, 3</td>
</tr>
<tr>
<td>1</td>
<td>2, 3</td>
</tr>
<tr>
<td>2</td>
<td>2, 3</td>
</tr>
<tr>
<td>3</td>
<td>3, 4</td>
</tr>
</tbody>
</table>
Challenge 4: adopting VGRAM

Easily adoptable by many algorithms

Basic interfaces:

- String s $\rightarrow$ grams
- String s1, s2 such that $ed(s1, s2) \leq k \rightarrow \min \# \text{ of their common grams}$
Lower bound on # of common grams

**Fixed length (q)**

If \( \text{ed}(s_1, s_2) \leq k \), then their # of common grams \( \geq \):

\[
(|s_1| - q + 1) - k \times q
\]

**Variable lengths:** # of grams of \( s_1 \) – \( \text{NAG}(s_1, k) \)
Example: algorithm using inverted lists

- Query: “shtick”, ED(shtick, ?)≤1

2-grams

<table>
<thead>
<tr>
<th>2-grams</th>
<th>2-4 grams</th>
</tr>
</thead>
<tbody>
<tr>
<td>ck</td>
<td>ck</td>
</tr>
<tr>
<td>ic</td>
<td>ic</td>
</tr>
<tr>
<td>ti</td>
<td>tic</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Lower bound = 3

<table>
<thead>
<tr>
<th>id</th>
<th>strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>rich</td>
</tr>
<tr>
<td>1</td>
<td>stick</td>
</tr>
<tr>
<td>2</td>
<td>stich</td>
</tr>
<tr>
<td>3</td>
<td>stuck</td>
</tr>
<tr>
<td>4</td>
<td>static</td>
</tr>
</tbody>
</table>

Lower bound = 1
End of part I

- Motivation
- Preliminaries
- Trie-based approach
- Gram-based algorithms
- Sketch-based algorithms
- Compression
- Selectivity estimation
- Transformations/Synonyms
- Conclusion

Part I

Part II