Similarity Search
The q-Gram Distance

Nikolaus Augsten
Free University of Bozen-Bolzano
Faculty of Computer Science
DIS
Unit 3 – March 15, 2012

Outline
1 Filters for the Edit Distance
   • Motivation
   • Lower Bound Filters
   • Length Filter
   • q-Grams: Count Filter
   • q-Grams: Position Filtering
   • Experiments
   • The q-Gram Distance

Application Scenario

Scenario:
• A company offers a number of services on the Web.
  - You can subscribe for each service independently.
  - Each service has its own database (no unique key across databases).

Example: customer tables of two different services:

<table>
<thead>
<tr>
<th></th>
<th>ID</th>
<th>name</th>
<th></th>
<th>ID</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1023</td>
<td>Frodo Baggins</td>
<td>...</td>
<td>948483</td>
<td>John R. R. Tolkien</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>J. R. R. Tolkien</td>
<td>...</td>
<td>153494</td>
<td>C. S. Lewis</td>
</tr>
<tr>
<td></td>
<td>239</td>
<td>C.S. Lewis</td>
<td>...</td>
<td>494392</td>
<td>Fordo Baggins</td>
</tr>
<tr>
<td></td>
<td>863</td>
<td>Bilbo Baggins</td>
<td>...</td>
<td>799294</td>
<td>Biblo Baggins</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Task: Created unified customer view!
Filters for the Edit Distance

Motivation

The Join Approach

- **Solution**: Join customer tables on name attribute (Q1):
  ```sql
  SELECT * FROM A, B
  WHERE A.name = B.name
  ```
- **Exact Join**: Does not work!
- **Similarity Join**: Allow $k$ errors...
  1. Register UDF (User Defined Function) for the edit distance:
     ```sql
     ed(x, y)
     ```
     returns the union cost edit distance between the strings $x$ and $y$.
  2. Rewrite query Q1 as similarity join (Q2):
     ```sql
     SELECT * FROM A, B
     WHERE ed(A.name, B.name) <= k
     ```

Effectiveness and Efficiency of the Approximate Join

Effectiveness:

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>ID</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1023</td>
<td>Frodo Baggins</td>
<td>494392</td>
<td>Fordo Baggins</td>
</tr>
<tr>
<td>21</td>
<td>J. R. R. Tolkien</td>
<td>948483</td>
<td>John R. R. Tolkien</td>
</tr>
<tr>
<td>239</td>
<td>C. S. Lewis</td>
<td>153494</td>
<td>C. S. Lewis</td>
</tr>
<tr>
<td>863</td>
<td>Bilbo Baggins</td>
<td>799294</td>
<td>Bilbo Baggins</td>
</tr>
</tbody>
</table>

⇒ very good (100% correct)

Efficiency: How does the DB evaluate the query?

1. compute $A \times B$
2. evaluate UDF on each tuple $t \in A \times B$

- **Experiment [GIJ+01]**: Self-join on string table (average string length = 14):
  - 1K tuples: ca. 30min
  - 14K tuples: > 3 days!

Prohibitive runtime!

Using a Filter for Search Space Reduction

- **Search space**: $A \times B$ (⇒ $|A| \cdot |B|$ edit distance computations)
- **Filtering (Pruning)**: Remove tuples that can not match, without actually computing the distance.

Outline

1. Filters for the Edit Distance
   - Motivation
   - Lower Bound Filters
     - Length Filter
     - $q$-Grams: Count Filter
     - $q$-Grams: Position Filtering
     - Experiments
     - The $q$-Gram Distance
Filters for the Edit Distance

Lower Bound Filters

- Lower bound (lb) for distance dist(x, y):
  \[ \text{dist}(x, y) \geq \text{lb}_{\text{dist}}(x, y) \]

- Query Q3 with Lower Bound Filter:
  ```sql
  SELECT * FROM A,B
  WHERE \text{lb}(A.name, B.name) \leq k AND
  \text{ed}(A.name, B.name) \leq k
  ```

  \text{lb}(A.name, B.name) is a cheap function

  Database will optimize query: compute \text{ed}(A.name, B.name) only if
  \[ \text{lb}(A.name, B.name) > k \]

  No false negatives!

Length Filtering

Theorem (Length Filtering [GIJ+01])

If two strings \(x\) and \(y\) are within edit distance \(k\), their lengths cannot differ by more than \(k\):

\[ \text{ed}(x, y) \geq \text{abs}(|x| - |y|) \]

- Proof: At least \(\text{abs}(|x| - |y|)\) inserts are needed to bring \(x\) and \(y\) to the same length.

- Query Q4 with Length Filtering:
  ```sql
  SELECT * FROM A,B
  WHERE \text{ABS}(\text{LENGTH}(A.name)-\text{LENGTH}(B.name)) \leq k AND
  \text{ed}(A.name, B.name) \leq k
  ```
Example: Length Filtering

- Execute query without/with length filter ($k = 3$):
  
<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>ID</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1023</td>
<td>Frodo Baggins13</td>
<td>948483</td>
<td>John R. Tolkien19</td>
</tr>
<tr>
<td>21</td>
<td>J. R. Tolkien16</td>
<td>153494</td>
<td>C. S. Lewis11</td>
</tr>
<tr>
<td>239</td>
<td>C. S. Lewis10</td>
<td>494392</td>
<td>Fordo Baggins13</td>
</tr>
<tr>
<td>863</td>
<td>Bilbo Baggins13</td>
<td>799294</td>
<td>Bilbo Baggins13</td>
</tr>
</tbody>
</table>

- Without length filter: 16 edit distance computations
- With length filter ($k = 3$): 12 edit distance computations
  
  J. R. Tolkien $\leftrightarrow$ C. S. Lewis is pruned
  all pairs (..., John R. Tolkien) except
  (J. R. Tolkien, John R. Tolkien) are pruned

What is a $q$-Gram?

- **Intuition:**
  - slide window of length $q$ over string $x \in \Sigma^*$
  - characters covered by window form a $q$-gram
  - where window extends string: fill with dummy character # $\notin \Sigma$

  **Example:** $x = \text{Frodo}$, $q = 3$

  extended: $$\text{F} \text{r} \text{o} \text{d} \text{o}$$

  $q$-grams:
  $$\text{# # F}$$
  $$\text{# F}$$
  $$\text{Fr}$$
  $$\text{Fro}$$
  $$\text{rod}$$
  $$\text{odo}$$
  $$\text{o #}$$

  $q$-Gram Profile $G_x$: bag of all $q$-grams of $x$

  Profile size: $|G_x| = |x| + q - 1$

Single Edit Operations and Changing $q$-Grams

- **Intuition:** Strings within small edit distance share many $q$-grams.
- How many $q$-grams ($q = 3$) change/remain?

  $ed(x, y) = 1 \Rightarrow |G_x \cap G_y| = \max(|G_x|, |G_y|) - q$
Multiple Edit Operations and Changing q-Grams

\[ \text{ed}(x, y) = 1 \Rightarrow |G_x \cap G_y| = \max(|G_x|, |G_y|) - q \]

What if ed(x, y) = k > 1?

| x   | |G_x| | y   | |G_y| | |G_x \cap G_y|
|-----|-----|-----|-----|-----|-----|-----|
| peter | 7   | meters | 8   | 2   |     |
| petal | 7   |   | 7   | 3   |     |

- Multiple edit operations may affect the same q-gram:
  - peter → G_x = \{##p, #pe, pet, etc., ter, er#, r##\}
  - petal → G_x = \{##p, #pe, pet, etc., tal, al#, l##\}

- Each edit operation affects at most q q-grams.

Count Filtering Query

Query Q5 with Count Filtering:

\[
\text{SELECT A.id, B.id, A.name, B.name}
\text{FROM A, QA, B, QB}
\text{WHERE A.id = QA.id AND B.id = QB.id AND QA.qgram = QB.qgram AND}
\text{ABS(LENGTH(A.name)-LENGTH(B.name)) <= k}
\text{GROUP BY A.id, B.id, A.name, B.name}
\text{HAVING COUNT(*) >= LENGTH(A.name)-1-(k-1)*q AND COUNT(*) >= LENGTH(B.name)-1-(k-1)*q AND}
\text{ed(A.name, B.name) <= k}
\]
Filters for the Edit Distance

### q-Grams: Count Filter

Problem with Count Filtering Query

- Previous query Q5 works fine for $kq < \max(|G_x|, |G_y|)$.
- However: If $kq \geq \max(|G_x|, |G_y|)$, no q-grams may match even if $\text{ed}(x, y) \leq k$.
- Example ($q = 3, k = 2$):
  
  WHERE-clause prunes $x$ and $y$, although $\text{ed}(x, y) \leq k$.

```
WHERE (x = 'IBM') AND (y = 'BMW')
```

x = IBM $G_x = \{##I, #IB, IBM, BM#, M##\}$ | $|G_x| = 5$

y = BMW $G_y = \{##B, #BM, BMW, MW#, W##\}$ | $|G_y| = 5$

- False negatives:
  - short strings with respect to edit distance (e.g., $|x| = 3, k = 3$)
  - even if within given edit distance, matches tend to be meaningless (e.g., abc and xyz are within edit distance $k = 3$)

Fixing Count Filtering Query

```
WHERE (LENGTH(A.name)+q-1 <= k*q AND LENGTH(B.name)+q-1 <= k*q AND ABS(LENGTH(A.name) - LENGTH(B.name)) <= k AND ed(A.name,B.name) <= k)
```

Note: We omit this part in subsequent versions of the query since it remains unchanged.

Outline

- Filters for the Edit Distance
  - Motivation
  - Lower Bound Filters
  - Length Filter
  - q-Grams: Count Filter
  - q-Grams: Position Filtering
  - Experiments
  - The q-Gram Distance

Positional q-Grams

- Enrich q-grams with position information:
  - extended string: prefix and suffix string $x$ with $q - 1$ characters #
  - slide window of length $q$ over extended string $x'$
  - characters covered by window after shifting it $i$ times form the q-gram at position $i + 1$

```
Example: x = Frodo
```

```
extended string:  #F r o d o #
```

```
positional q-grams:
(1, #F)  (2, #Fr)  (3, #Fro)  (4, #rod)  (5, #odo)  (6, #odo)  (7, #o#)
```
Given: table \( N \)
- \( N \) has a single attribute \( i \)
- \( N \) is filled with numbers from 1 to \( max \)
  \( (max \) is the maximum string length plus \( q - 1 \))

Positional \( q \)-grams for table \( A \) in SQL (Q7):

```sql
CREATE TABLE QA AS
    SELECT A.id, N.i AS pos,
    SUBSTRING(CONCAT(
        SUBSTRING('#..#', 1, q - 1),
        LOWER(A.name),
        SUBSTRING('#..#', 1, q - 1)),
    N.i, q) AS qgram
    FROM A, N
    WHERE N.i <= LENGTH(A.name) + q - 1
```

Corresponding \( q \)-grams:
- Given: positional \( q \)-grams \((i, g)\) of \( x \)
- transform \( x \) to \( y \) applying edit operations
- \((i, g)\) “becomes” \((j, g)\) in \( y \)
- We define: \((i, g)\) corresponds to \((j, g)\)

Example:
- \( x' = ###abaZabaaba## \), \( y' = ###abaabaabaabaabababaa### \)
- edit distance is 1 (delete \( Z \) from \( x \))
- \((7, aba)\) in \( x \) corresponds to \((6, aba)\) in \( y \)
- …but not to \((9, aba)\)

Query Q8 with Count and Position Filtering:

```sql
SELECT A.id, B.id, A.name, B.name
FROM A, QA, B, QB
WHERE A.id = QA.id AND
    B.id = QB.id AND
    QA.qgram = QB.qgram AND
    ABS(LENGTH(A.name)-LENGTH(B.name)) <= k AND
    ABS(QA.pos-QB.pos)<=k
GROUP BY A.id, B.id, A.name, B.name
HAVING COUNT(*) >= LENGTH(A.name)-1-(k-1)*q AND
    COUNT(*) >= LENGTH(B.name)-1-(k-1)*q AND
    ed(A.name,B.name) <= k
```

Proof:
- each increment (decrement) of a position requires an insert (delete);
- a shift by \( k \) positions requires \( k \) inserts/deletes.
# Filters for the Edit Distance

## Motivation

## Lower Bound Filters

- Length Filter
- $q$-Grams: Count Filter
- $q$-Grams: Position Filtering

## Experiments

### The q-Gram Distance

- All experimental results taken from [GIJ+01]
- Three string data sets:
  - set1: 40K tuples, average length: 14 chars
  - set2: 30K tuples, average length: 38 chars
  - set3: 30K tuples, average length: 33 chars

## Experimental Data

### String Length Distributions

- Set 1
- Set 2
- Set 3

## Candidate Set Size

- Question: How many edit distances do we have to compute?
- Show candidate set size for different filters (small is good).
- $q = 2$
- Caption:
  - CP: cross product
  - L: length filtering, P: position filtering, C: count filtering
  - Real: number of real matches

- Set 1
### Filters for the Edit Distance Experiments

#### Candidate Set Size

- **Question:** How many edit distances do we have to compute?
- **Show candidate set size for different filters (small is good).**
- **q = 2**
- **Caption:**
  - CP: cross product
  - L: length filtering, P: position filtering, C: count filtering
  - Real: number of real matches

#### Various q-Gram Lengths

- **Question:** How does the choice of q influence the filter effectiveness?
- **Show candidate set size for different q values (small is good).**

#### Response Time

- **Approximate self-join on small sample of 1000 tuples (set 1)**
  - (full dataset > 3 days without filters!)
- **Measure response time (small is good).**
- **Caption:**
  - k: edit distance threshold
  - Q1: edit distance without filters
  - Q2: edit distance with filters
Filters for the Edit Distance

The q-Gram Distance

Definition (q-Gram Distance [Ukk92])

Let $G_x$ and $G_y$ be the q-gram profiles of the strings $x$ and $y$, respectively. The q-gram distance between two strings is the number of q-grams in $G_x$ and $G_y$ that have no match in the other profile,

$$\text{dist}_q(x, y) = |G_x \cup G_y| - 2|G_x \cap G_y|.$$

Example: $q = 2$, $x = \text{abab}$, $y = \text{abcab}$

$G_x = \{\#a, \#b, \text{ab}, \text{ba}, \text{ab}, \text{b}\}$

$G_y = \{\#a, \#b, \text{ab}, \text{bc}, \text{ca}, \text{ab}, \text{b}\}$

$G_x \cup G_y = \{\#a, \#b, \text{ab}, \text{bc}, \text{ca}, \text{ab}, \text{b}\}$

$G_x \cap G_y = \{\#a, \#b, \text{ab}, \text{b}\}$

$\text{dist}_q(x, y) = |G_x \cup G_y| - 2|G_x \cap G_y| = 11 - 2 \cdot 4 = 3$

Pseudo Metric q-Gram Distance

The q-gram distance is a pseudo metric:

- For all $x, y, z \in \Sigma^*$
  - $\text{dist}_q(x, y) + \text{dist}_q(y, z) \geq \text{dist}_q(x, z)$ (triangle inequality)
  - $\text{dist}_q(x, y) = \text{dist}_q(y, x)$ (symmetric)
  - $\text{dist}_q(x, y) = 0 \iff x = y$

Note: Identity condition relaxed: $\text{dist}_q(x, y) = 0 \iff x = y$

i.e., the q-gram distance between two different strings can be 0

Example:

$\text{dist}_q(\text{axybxy}, \text{axycxybxyd}) = 0$

$G_x = G_y = \{\#a, \#x, \text{axy}, \text{bxy}, \text{cxy}, \text{y}, \text{xy}, \text{c}, \text{y}, \text{d}, \text{##}\}$

Distance Normalization (1/3)

What is a good threshold?

$\text{ed(IBM, BMW)} = 2$

$\text{ed(International Business Machines Corporation)} = 17$

Problem: Absolute numbers not always meaningful...

Solution: Compute error relative to string length!
Distance Normalization (2/3)

- Normalize distance such that $\delta(x, y) \in [0..1]$
- **Edit Distance**: $0 \leq ed(x, y) \leq \max(|x|, |y|)$
- Normalized Edit Distance: $0 \leq \text{norm-ed}(x, y) \leq 1$
  
  $$\text{norm-ed}(x, y) = \frac{ed(x, y)}{\max(|x|, |y|)}$$

- **q-Gram Distance**: $0 \leq \text{dist}_q(x, y) \leq |G_x \cup G_y| - |G_x \cap G_y|$
- Normalized q-Gram Distance: $0 \leq \text{norm-dist}_q(x, y) \leq 1$
  
  $$\text{norm-dist}_q(x, y) = \frac{\text{dist}_q(x, y)}{|G_x \cup G_y| - |G_x \cap G_y|} = 1 - \frac{|G_x \cap G_y|}{|G_x \cup G_y| - |G_x \cap G_y|}$$

Dividing by $|G_x \cup G_y|$ also normalizes to $[0..1]$, but the metric properties (triangle inequality) get lost [ABG10].

Distance Normalization (3/3)

- Normalized edit distance:
  
  $$\text{norm-ed}(\text{International Business Machines Corporation}, \text{International Business Machine Corporation}) = 0.047$$
  $$\text{norm-ed}(\text{IBM, BMW}) = 0.66$$
  $$\text{norm-ed}(\text{Int. Business Machines Corp., International Business Machines Corporation}) = 0.4$$

- Normalized q-gram distance ($q = 3$):
  
  $$\text{norm-dist}_q(\text{International Business Machines Corporation}, \text{International Business Machine Corporation}) = 0.089$$
  $$\text{norm-dist}_q(\text{IBM, BMW}) = 1.0$$
  $$\text{norm-dist}_q(\text{Int. Business Machines Corp., International Business Machines Corporation}) = 0.36$$

Edit Distance vs. q-Gram Distance

- Edit distance can not handle block-moves well:
  
  $x = \text{Nikolaus Augsten}$  $y = \text{Augsten Nikolaus}$
  
  norm-ed$(x, y) = 1.0$
  
  norm-dist$_q(x, y) = 0.39$  ($q = 3$)

- q-Gram distance may be too strict:
  
  $x = +39-06-46-74-22$  $y = (39 \ 06 \ 467422)$
  
  norm-ed$(x, y) = 0.4$
  
  norm-dist$_q(x, y) = 1.0$  ($q = 3$)