Approximate String Matching

Yufei Tao

KAIST

May 21, 2013

In this lecture, we will discuss the following approximate string matching problem.

Problem (Approximate String Matching)

Let S be a set of strings, each of which has a distinct id. Given a query string t and a value d, we want to report the ids of all strings in S that are within edit distance d from t.

This is a key problem for search engines to provide automatic correction of spelling errors. In practice, d is a small value, e.g., 1.

Example

Suppose that S has $\{\mathtt{ab}, \mathtt{abbb}, \mathtt{babb}, \mathtt{bbaa}, \mathtt{bbbaa}\}$, where the strings have ids 1 to 5 (from left to right). Then:

- for $t = \mathtt{bba}$ and $t = 1$, the query returns 4.
- for $t = \mathtt{bab}$ and $t = 2$, the query returns 1, 2, 3, and 4.

We will consider the problem in the scenario where $|t| > d$. As mentioned, a typical value of d in practice is 1, in which case essentially we require $|t| \geq 2$.

We will discuss some solutions that have been confirmed to be highly effective in practice. They, however, are heuristic in the sense that they do not have attractive worst case performance guarantees in the worst case.

If

- the query string t is short
- the alphabet has a small size, and
- d is small

we can convert the problem into exact matching as follows. Generate all the d-neighbors of t, i.e., strings that are within edit distance d from t.

Example

For $t = \mathtt{ab}$ and a binary alphabet $\{\mathtt{a}, \mathtt{b}\}$, all the 1-neighbors of t include $\{\mathtt{a}, \mathtt{b}, \mathtt{bb}, \mathtt{aa}, \mathtt{aab}, \mathtt{bab}, \mathtt{abb}, \mathtt{aba}\}$.

Observe that, for each $s \in S$, $edit(s, t) \leq d$ if and only if s is a d-neighbor of t. Hence, we can answer the approximate matching query by finding the exact matches of all d-neighbors of t.

Think: What is the most serious drawback of this solution?

Next, we discuss solutions that are more efficient in practice.

As before, we will denote by $s[i]$ the i-th letter of a string s ($1 \leq i \leq |s|$), and by $s[i..j]$ ($1 \leq i \leq j \leq |s|$) the substring of s that starts at $s[i]$ and ends at $s[j]$. Also, let $\#$ be a special symbol that does not exist in any data string of S.

Definition (q-gram)

Let s be a string, and (i, σ) be a pair where i is an integer and σ a string of length q. Then, (i, t) is a positional q-gram of s if either of the following conditions holds:

- $i \leq |s| - q + 1$ and $\sigma = s[i..i + q - 1]$
- $i \in [|s| - q + 1, |s|]$, and σ is formed by concatenating $s[i..|s|]$ and a sequence of $\#$.

If (i, σ) is a positional q-gram, then σ is a q-gram of s.

Example

Consider $s = \texttt{bbaa}$. Then, s has

- four positional 1-grams: $(1, \texttt{b}), (2, \texttt{b}), (3, \texttt{a}), (4, \texttt{a})$.
- two 1-grams: $\texttt{a}$ and $\texttt{b}$.
- four positional 2-grams: $(1, \texttt{bb}), (2, \texttt{ba}), (3, \texttt{aa}), (4, \texttt{a\#})$.
- four 2-grams: $\texttt{bb}, \texttt{ba}, \texttt{aa}, \texttt{a\#}$.

We will denote by $Q_q(s)$ the set of all positional q-grams of s. Clearly, $|Q_q(s)| = |s|$.

Let s and t be two strings. Let (i, σ_s) and (j, σ_t) be positional q-grams of s and t, respectively. We say that (i, σ_s) d-matches (j, σ_t) if:

- $\sigma_s = \sigma_t$
- $|i - j| \leq d$.

We will also say that (i, σ_s) is a d-match of (j, σ_t), and conversely, (j, σ_t) is a d-match of (i, σ_s).

Lemma

If $edit(s, t) \leq d$, then $Q_q(t)$ has at least

$$\max\{|s|, |t|\} - dq$$

positional q-grams each having at least a d-match in $Q_q(s)$.

Example

Suppose that $s = \mathtt{bbaa}$. Let $q = 2$ and $d = 1$.

- For $t = \mathtt{bba}$, $Q_q(t)$ has 3 positional q-grams that have d-matches in $Q_q(s)$: $(1, \mathtt{bb}), (2, \mathtt{ba}), (3, \mathtt{a\#})$.
- For $t = \mathtt{ba}$, $Q_q(t)$ has 1 positional q-gram with a d-match in $Q_q(s)$: $(1, \mathtt{ba})$.
- For $t = \mathtt{aa}$, $Q_q(t)$ has no positional q-gram with a d-match in $Q_q(s)$.

Note that the reverse of the lemma is not true. Namely, even if $Q_q(t)$ has at least $\max\{|s|, |t|\} - dq$ positional q-grams with d-matches in $Q_q(s)$, it is still possible that $edit(s, t) > d$.

Example

Suppose that $s = \mathtt{bbaa}$. Let $q = 2$ and $d = 1$.

- For $t = \mathtt{bbb}$, $Q_q(t)$ has 2 positional q-grams that have d-matches in $Q_q(s)$: $(1, \mathtt{bb}), (2, \mathtt{bb})$.

The previous lemma indicates a filter refinement strategy for solving the approximate string matching problem:

1. (Filter) Find the set C of candidate strings $s \in S$ such that $Q_q(t)$ has at least $|t| - dq$ positional q-grams with d-matches in $Q_q(s)$.
2. (Refinement) For every candidate string $s \in C$, verify whether $edit(s, t) \leq d$.

We focus on the filter step because the refinement step simply invokes the edit distance verification algorithm we discussed before.

We will instead focus on the following q-gram matching problem. Given a positional q-gram (i, σ), find all the strings in $s \in S$ such that (i, σ) has at least a d-match in $Q_q(s)$.

Think

How do we implement the filter step efficiently by leveraging a solution to the above problem?

We will solve the q-gram matching problem using an inverted index, as shown next.

Let Q be the set of q-grams of all the strings in S. We create an inverted index as follows.

- For each q-gram $\sigma \in Q$, create an inverted list $list(\sigma)$.
- For every string $s \in S$ with a position q-gram (i, σ) for some $i \in [1, |s|]$, the inverted list $list(\sigma)$ contains an entry $(id(s), i)$, where $id(s)$ is the id of s.

Example

Suppose that S has $\{\mathtt{ab}, \mathtt{abbb}, \mathtt{babb}, \mathtt{bbaa}, \mathtt{bbbaa}\}$, where the strings have ids 1 to 5 (from left to right). Let $q = 2$. The inverted index is:

q-gram σ	inverted list for σ
aa	(4, 3), (5, 4)
ab	(1, 1), (2, 1), (3, 2)
a#	(4, 4), (5, 5)
ba	(3, 1), (4, 2), (5, 3)
bb	(2, 2), (2, 3), (3, 3), (4, 1), (5, 1), (5, 2)
b#	(1, 2), (2, 4), (3, 4)

Think

- How would you use the above index to find the ids of the strings s such that $Q_2(s)$ has a 1-match of $(3, \mathtt{ab})$?
- Putting everything together, how would answer a query with $t = \mathtt{bbab}$ and $d = 1$?