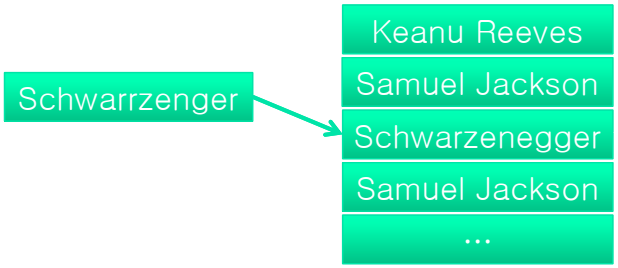
VGRAM

C. Li, B. Wang, X. Yang: VLDB 2007

Presented by Kyuseok Shim

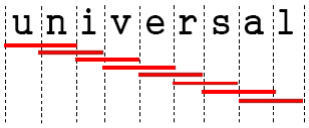
Approximate selection queries

- Spell checking
- Query relaxation



"Q-Grams" of strings

- Example
  - 2-grams



Q-Gram inverted lists

- Inverted Index
  - Posting list: for a q-gram, the sequence of record id it appears
  - Posting: the element of posting list

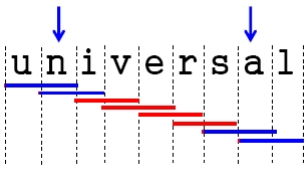
id	strings
0	rich
1	stick
2	stich
3	stuck
4	static

2-grams

at	4
ch	0 → 2
ck	1 → 3
ic	0 → 1 → 2 → 4
ri	0
st	1 → 2 → 3 → 4
ta	4
ti	1 → 2 → 4
tu	3
uc	3

Edit operation's effect on grams

- k operations could affect k*q grams



  - E.g.,
    - Operation on 'n' at position 2 effect to "un, ni"
    - Operation on 's' at position 8 effect to "sa, al"

Minimum Common Grams

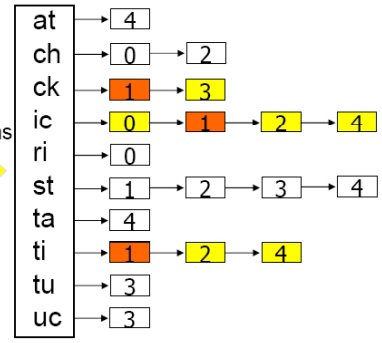
- Given,
  - s_1, s_2: two strings
  - q: the length of gram
  - k: maximum edit distance
- The two set of grams of s_1, s_2 should share at least,
  - $B_c(s_1, s_2, q, k) = \max\{|s_1|, |s_2|\} - q+1 - k*q$

Searching using inverted lists

- Query: shtick, ED(shtick, ?) ≤ 1
  - Q-grams: sh ht ti ic ck
  - # of common grams is alt least 3

id	strings
0	rich
1	stick
2	stich
3	stuck
4	static

2-grams



Records Sharing q-grams

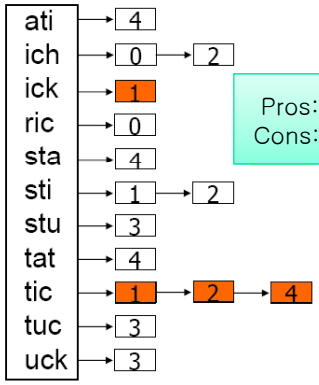
- How to find the records that share at least k q-grams?
  - To join for all case of subset of q-grams
  - E.g., for a query string 'string', q=3 and k=3
    - Q-gram set: {str, tri, rin, ing}
    - All possible subsets to share 3 q-grams
      {str, tri, rig}
      {str, tri, ing}
      {str, rin, ing}
      {tri, rin, ing}
  - Union for all the result of join of possible q-gram subsets
- Counting method
  - While querying for all q-grams in query string, counting for the record id appears in posting list

2-grams → 3-grams ?

- Query: shtick, ED(shtick, ?) ≤ 1
  - Q-grams: sht hti tic ick

id	strings
0	rich
1	stick
2	stich
3	stuck
4	static

3-grams



Pros: Shorter inverted list
Cons: More false positives

Motivation

- Small index size in memory
- Small running time

Variable length grams?

id	string
1	bingo
2	bioinng
3	bitingin
4	biting
5	boing
6	going

gram		string ids
bi	→	1, 2, 3, 4
bo	→	5
gi	→	3
go	→	1, 6
in	→	1, 2, 3, 3, 4, 5, 6
io	→	2
it	→	3, 4
ng	→	1, 2, 3, 4, 5, 6
nn	→	2
oi	→	2, 5, 6
ti	→	3, 4

Size: 30

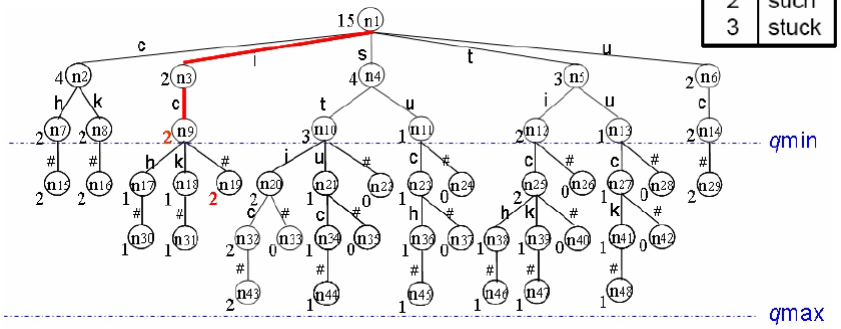
gram		string ids
bi	→	1, 2, 3, 4
bo	→	5
gi	→	3
go	→	1, 6
in	→	2, 3
ing	→	1, 3, 4, 5, 6
io	→	2
it	→	3, 4
ng	→	2
nn	→	2
oi	→	2, 5, 6
ti	→	3, 4

Size: 25

Constructing vgram dictionary

- Prune trie using a frequency threshold T (e.g., 2)

id	string
0	stick
1	stich
2	such
3	stuck

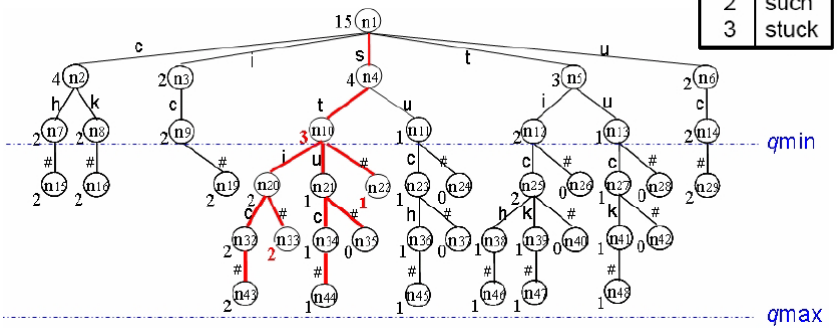


A gram-frequency trie: [2,4]-gram

Constructing vgram dictionary

- Prune trie using a frequency threshold T (e.g., 2)

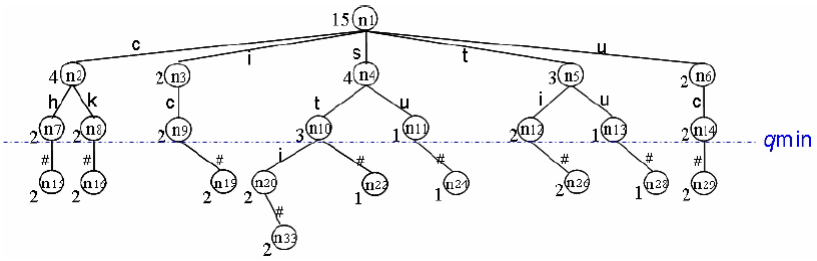
id	string
0	stick
1	stich
2	such
3	stuck



A gram-frequency trie: [2,4]-gram

Final vgram dictionary

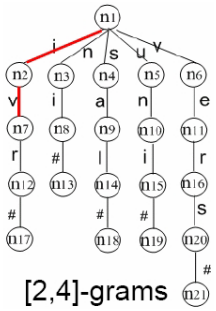
- The result trie



Generating q-grams for a query string

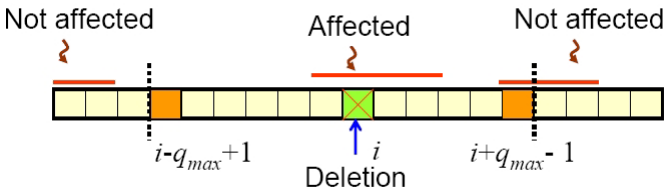
- Generating q-grams for a string with the dictionary
  - Select the longest q-gram in the dictionary
  - E.g., with the dictionary of right figure, for a string 'universal'

u n i v e r s a l



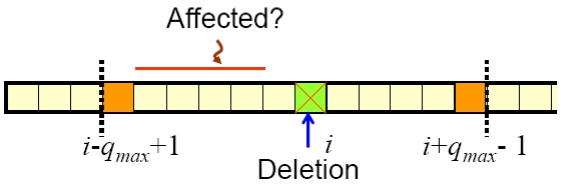
Deletion affects variable-length grams

- Do not affect to the grams ending before $i-q_{max}+1$
- Do not affect to the grams starting after $i+q_{max}-1$
- Affect to overlapping grams with the deleted position



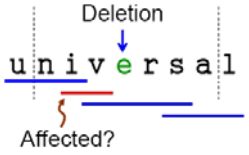
Vgrams affected by a deletion

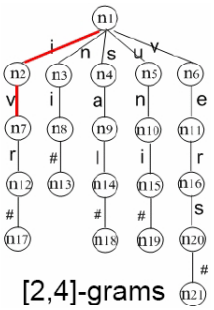
- Not sure about the grams ending after $i-q_{max}+1$ but before the deletion
- Not sure about the grams starting before $i+q_{max}+1$ but after the deletion



Vgrams affected by a deletion

- 'iv' is affected if there is a gram that have 'iv' as prefix in dictionary

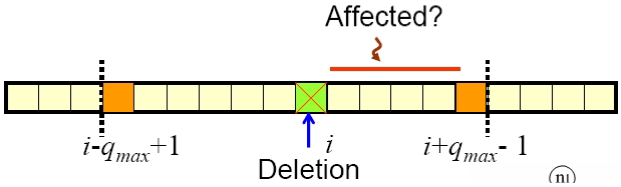


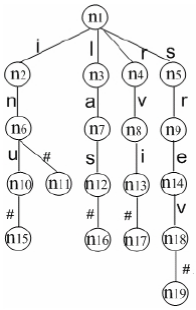


- The edited string 'univrsal'
  - Q-grams: uni, ivr, rs, sal
  - it is an answer to the query ED('universal', 1)≤1
  - The 'universal' has 2 non-affected gram: {uni, sal}, and 'univrsal' is share at least 2 gram, so 'univrsal' can be answered

Con't

- A gram ending before $i+q_{max}+1$ and after the deletion is affected if there is some grams that have it as a suffix
- Reverse trie





Trie of reversed grams

Pre-calculate # of affected grams

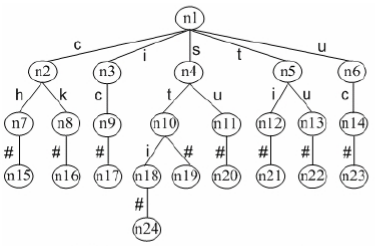
- Pre-calculate the maximum number of grams affected
  - E.g., universal → <2, 4>
    - With 1 edit operations, at most 2 grams affected
    - With 2 edit operations, at most 4 grams affected
- Called NAG vector
  - Number of Affected Grams
- For a string s_i, let $VG(s_i)$ and $NAG(s_i)$ be the corresponding set of vq-grams and NAG vector of s_i, respectively
  - For two string, s_1 and s_2, $ed(s_1, s_2) \leq k$
  - Lower bound on the # of common g-gram
  - $Bvc(s_1,s_2,k) = \max \{ |VG(s_1)| - NAG(s_1,k), |VG(s_2)| - NAG(s_2,k) \}$

Summary

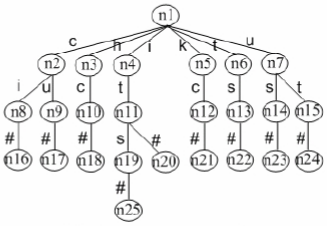
id	string
0	stick
1	stich
2	such
3	stuck

{sti, ic, ck}

(a) strings



(b) Gram dictionary as a trie



(c) Reversed-gram trie

id	NAG vector
0	2, 3
1	2, 3
2	2, 3
3	3, 4

(d) NAG vectors