q-gram

- un *q-gram* est simplement une suite de q lettres dans un texte

q-gram

- un *q-gram* est simplement une suite de q lettres dans un texte
- une *lettre* c'est juste un item dans un alphabet : on peut aussi voir un *q-gram* comme une suite de q lexèmes

Exemple

- un texte

```
A do run run run, a do run run
```

Exemple

- un texte

  `A do run run run, a do run run`

- le texte nettoyé

  `adorunrunrunadorunrun`

Exemple

- un texte

  `A do run run run, a do run run`

- le texte nettoyé

  `adorunrunrunadorunrun`

- la transformation en *q-grams*

  `adoru dorun orunr runru unrun nrunr runru`
  `unrun nruna runad unado nador adoru dorun`
  `orunr runru unrun`

Analyse en fréquence

46	[identifier]	40	[identifier]	30	[identifier]
3	[stringliteral]	9	[integerliteral]	7	[integerliteral]
3	[new]	3	[stringliteral]	2	[stringliteral]
3	[integerliteral]	2	[public]	2	[public]
2	[public]	2	[import]	2	[import]
2	[import]	1	[void]	1	[void]
1	[void]	1	[static]	1	[static]
1	[static]	1	[package]	1	[package]
1	[package]	1	[new]	1	[new]
1	[class]	1	[class]	1	[class]

Analyse en fréquence

```
23 [id, id, id]
 3 [new, id, id]
 3 [int, id, id]
 3 [id, new, id]
 3 [id, int, id]
 3 [id, id, str]
 3 [id, id, new]
 3 [id, id, int]
 2 [str, id, id]
 2 [id, str, id]
 2 [id, import, id]
```

```
21 [id, id, id]
 5 [int, id, id]
 3 [str, int, int]
 3 [id, str, int]
 3 [id, int, id]
 3 [id, id, str]
 3 [id, id, int]
 2 [int, int, id]
 2 [id, import, id]
```

```
17 [id, id, id]
 3 [id, id, int]
 2 [str, int, int]
 2 [int, str, int]
 2 [int, id, id]
 2 [id, int, str]
 2 [id, import, id]
```

q-gram

- un *q-gram* est simplement une suite de q lettres dans un texte
- une *lettre* c'est juste un item dans un alphabet : on peut aussi voir un *q-gram* comme une suite de q lexèmes
- on peut associer un entier à chaque *q-gram* : disposer d'une fonction de hachage

Exemple

- un texte

  `A do run run run, a do run run`

- le texte nettoyé

  `adorunrunrunadorunrun`

- la transformation en *q-grams*

  `adoru dorun orunr runru unrun nrunr runru unrun nruna runad unado nador adoru dorun orunr runru unrun`

- les *hashes* des *q-gram*

  `77 72 42 17 98 50 17 98 8 88 67 39 77 72 42 17 98`

Fingerprint

Pourquoi ?

Calculer une *empreinte* permettant de s'affranchir de la comparaison de tous les *q-gram*.

Comment ?

- calculer un sous-ensemble des *q-gram* ?
- lequel ?
- quelle propriété ?

Approche de Manber

Udi Manber. *Finding similar files in a large file system*. In Proceedings of the USENIX Winter 1994 Technical Conference, pages 1–10, San Fransisco, CA, USA, 17–21 1994

On conserve les h hashes des *q-gram* tels que $h \mod p = 0$ pour un p donné.

Exemple

- un texte

  `A do run run run, a do run run`

- le texte nettoyé

  `adorunrunrunadorunrun`

- la transformation en *q-grams*

  `adoru dorun orunr runru unrun nrunr runru unrun nruna runad unado nador adoru dorun orunr runru unrun`

- les *hashes* des *q-gram*

  `77 72 42 17 98 50 17 98 8 88 67 39 77 72 42 17 98`

- les *hashes* tels que $h \mod 4 = 0$

  `72 8 88 72`

Approche de Heintze

Nevin Heintze. *Scalable document fingerprinting*. In 1996 USENIX Workshop on Electronic Commerce, November 1996.

On conserve les hashes des n plus petites valeurs de hash des *q-gram*.

Winnowing

Schleimer ET AL.. *Winnowing : Local Algorithms for Document Fingerprinting*. In SIGMOD '03 Proceedings of the 2003 ACM SIGMOD international conference on Management of data, 2003.

Objectif

- seuil de garantie t : si il existe un facteur commun de longueur $\geq t$ alors il est détecté
- seuil de bruit k : tout facteur plus court que k ne doit pas être détecté

Winnowing

Schleimer ET AL.. *Winnowing : Local Algorithms for Document Fingerprinting*. In SIGMOD '03 Proceedings of the 2003 ACM SIGMOD international conference on Management of data, 2003.

Objectif

- seuil de garantie t : si il existe un facteur commun de longueur $\geq t$ alors il est détecté
- seuil de bruit k : tout facteur plus court que k ne doit pas être détecté

Proposition

Dans chaque fenêtre, sélectionner la valeur de hash minimale. Si il y en a plus d'une occurrence, sélectionner celle dont la position est la plus à droite dans sa fenêtre.

Exemple

- les *hashes* des *q-gram*

```
77 72 42 17 98 50 17 98 8 88 67 39 77 72
42 17 98
```

Exemple

- les *hashes* des *q-gram*

  77 72 42 17 98 50 17 98 8 88 67 39 77 72
  42 17 98

- fenêtres de taille 4

  (77, 74, 42, **17**) (74, 42, 17, 98)
  (42, 17, 98, 50) (17, 98, 50, **17**)
  (98, 50, 17, 98) (50, 17, 98, **8**)
  (17, 98, 8, 88) (98, 8, 88, 67)
  (8, 88, 67, 39) (88, 67, **39**, 77)
  (67, 39, 77, 74) (39, 77, 74, 42)
  (77, 74, 42, **17**) (74, 42, 17, 98)

Exemple

- les *hashes* des *q-gram*

  77 72 42 17 98 50 17 98 8 88 67 39 77 72
  42 17 98

- fenêtres de taille 4

  (77, 74, 42, **17**) (74, 42, 17, 98)
  (42, 17, 98, 50) (17, 98, 50, **17**)
  (98, 50, 17, 98) (50, 17, 98, **8**)
  (17, 98, 8, 88) (98, 8, 88, 67)
  (8, 88, 67, 39) (88, 67, **39**, 77)
  (67, 39, 77, 74) (39, 77, 74, 42)
  (77, 74, 42, **17**) (74, 42, 17, 98)

- empreinte

  17 17 8 39 17

  [17,3] [17,6] [8,8] [39,11] [17,15]

Algorithme de Karp-Rabin

Richard M. Karp and Michael O. Rabin. Efficient randomized pattern-matching algorithms. IBM Journal of Research and Development, 31(2) :249–260, 1987.

Objectif

Rechercher un motif m dans un texte t.

- quelle complexité ?
- comment réutiliser les notions vues avant ? ▸ Exemple