Stop lists.

Motivation & Methods of Stemming

Calculating TF-IDF Simularity

- Stop words Removal : Remove 'Noise words' from text
  Contribute no info. to info. Retreival Process.

find examples

- Stemming : Remove irrelavent differences.
  Different forms of the same word.

- Semantic Relationships.

Stemming (Morphology)

- A query. & a doc. contain different forms of the same word.
  Related.
- Remove surface markings.
  Reveal basic form.

form → forms
stem → forming
→ former
⋮

- Replace words with its eq. classes of words
  stems

⇒ Reduce number of different words
Increase the number of instances of each token.

- Question raised : Not all words obey regular rules
- Solution : Identify sub-pattern of letters
  ⇒ devise rules to deal with patterns

Stemmer : implements stemming algorithm.
Using stemer ⇒ Reduce Vocab. size 10%~50%

Stop words "Noise" : vital for grammar
useless for idetitifying the content

- Speified in a text file Stop list

Matching Query q. Document d.

Sim(q,d) ⇒ def. Simularity

- No. terms common to q and d.
- how useful. is to common term.
  e.g. "the" and "mgnesium"

IDF weighting : Measuring Significance

inverse document frequency

$$IDF(t) = \log\left(\frac{\overset{\text{total num. of Docs.}}{ND}}{\underset{\text{\#. Docs include t.}}{ND_t}}\right)$$

- case ① t occurs in every docs — $ND = ND_t$. $\log(1) = 0$

  ② t in a few docs — $ND > ND_t$. $\log\left(\frac{ND}{ND_t}\right) > 0$.

* ignore the occurence frequency within each doc.

slide 22. (week4)

Document length.

TF-IDF weight

term frequency - Inverse Document frequency

$$\underline{\underline{w_{td}}} = f_{td} \cdot IDF(t)$$

(Number of times t occurs in d) · Inverse $\overset{\text{doc.}}{\vee}$ frequency of t.

f_{td} Large: often occur in d.

IDF(t) large: occur in few docs

Query weights

- long query $\underline{q}$ treat as document

  $w_{tq} = \overline{f_{tq}} \cdot IDF(t)$

  # times t in q

- Short query $\underline{q}$

  $w_{tq} = IDF(t)$

<u>TF-IDF Similarity</u> between q and d.

$$Sim(q.d) = \frac{\sum_{t \in q \cap d} w_{td} \cdot w_{tq}}{\|d\| \cdot \|q\|}$$

For every t in both q and d.

- Calculate document TF-IDF weight
  Calculate query weight } <u>Sum the product</u>

Document length

$$\text{Len}(d) = \|d\| = \sqrt{\sum_{t \in d} w_{td}^2}$$

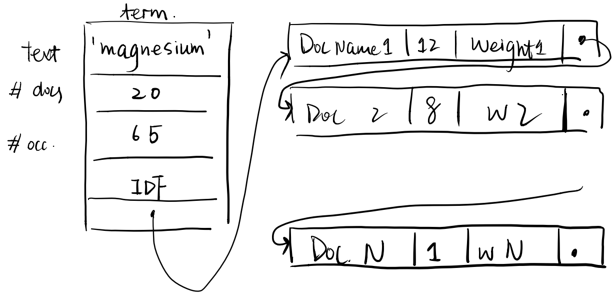
$f_{td} \cdot IDF(t)$

vector space

$$\|X\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

Document Index

Document Index: Speed up computation of Sim(q,d.)



?

~~Docs are ordered by the occurence of t~~

- Practical:
  - terms: ordered in decreasing IDF

"unique"

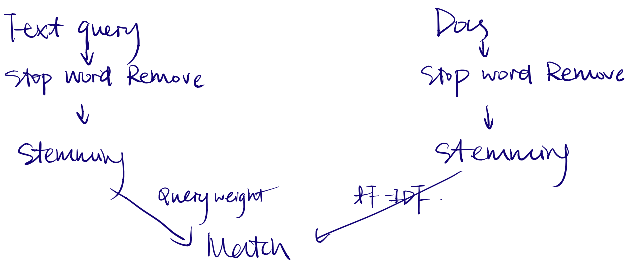
- For each term: Docs: decreasing weight

"if for t, its often and unique" ⇒ weights higher.

- For each term in Query
  - identify ~ in index.
  - Increment similarity scores → 相加
  - Stop when weights fall bellow threshold.

While calculating Similarity

IR Process



Summary of Points

- Stop Lists
- Stemming
- TF-IDF weight for Docs
- Query weight
- "Length" of doc. Len.(d)
- TF-IDF similarity
- Document index. ⇒ speedup Sim. calculation.