Date

* Weighted Zone Scoring:

- Document is divided into zones
- For a query q and a document d, weighted zone scoring assigns to pair (q,d) a score in range [0,1] by computing a linear combination of zone scores
- For a set of documents, each document has l zones.
- Let $g_1, \dots, g_L \in [0,1]$ such that $\sum_{i=1}^{L} g_i = 1$
- $\forall_{i=1}^{L}$, s_i be the boolean score denoting a match between q and i^{th} zone

  $s_i = 1$, if all query term occur in that zone

  otherwise $s_i = 0$.
- Weighted zone score $\Rightarrow \sum_{i=1}^{L} g_i \times s_i$.

* Learning weights for zone scoring:

- How to determine weights g_i for weighted zone scoring?
- These weights are learned using training examples that have been judged editorially.

- There is a set of training examples each of which is a tuple of a query q, a document d and a relevance judgement for d on q.

Chap #06

①

Scoring, Term Weighting & Vector Space Model.

Date

Ranked Retrieval	Unranked Retrieval.
① System retrieves document wrt ranking order.	① System retrieves flat result with no ranking.
② Assign scores to each term for matching.	② Binary criterion for deciding relevance.
③ Supports free text queries as well as boolean queries.	③ Info need has to be translated in to boolean queries.

* Parametric Search :

→ Documents contain
- Data
- Meta data ⇒ specific form data associated with each document.

eg: author of book, date of publication etc

→ Provides search based on parameter.

→ Parametric search consists as usual of postings intersection and we can merge postings by standard inverted indexes as well as parametric indexes.

S_T	S_B	Score
0	0	0
0	1	$1-g$
1	0	g
1	1	1

Date

score(d, q) =
$g \cdot S_T(d, q) + (1-g) S_B(d, q)$

$S_T \Rightarrow g_t$	title	(g)
$S_B \Rightarrow g_B$	body	$(1-g)$

error of scoring function with weight g

$$\varepsilon(g, \phi_j) = \left(r(d_j, q_j) - \text{score}(d_j, q_j)\right)^2$$

where r = editorial relevance judgement quantized to 0, 1

Total error = $\sum_j \varepsilon(g, \phi_j)$.

eg Training examples n_{01} = relevant,
n_{02} = irrelevant
$S_T = 0$, $S_B = 1$

error = $(r(d, q) - s(d, q))$

n_{01} error = $[1 - (1-g)]^2 n_{01}$

n_{02} error = $[0 - (1-g)] n_{02}$

Total error = $[1 - (1-g)]^2 n_{01} + [0 - (1-g)]^2 n_{02}$

nice

Date

Q When using weighted zone scoring, is it necessary for all zones to use same Boolean function?

Ans No,
Boolean score for title zone could be 1 when atleast half of the query terms occur in the zone and 0 otherwise.
Boolean score for body zone could be 1 when all query terms occur in the body & 0 otherwise.

Q Author zone $g_1 = 0.2$, title zone $g_2 = 0.31$, body zone $g_3 = 0.49$.
Distinct scores?

Ans 1 if appears in all zones.
0.51 if appears in author & title zone.
0.69 " " " & body zone
0.8 " " " title & " " .

* Term frequency:
  - No. of occurences of term t in document
  - How many times term appear in a document?
  - Denoted by $tf_{t,d}$

Date

* Document Frequency:
- No. of documents that contains term t.
- Denoted by df_t.

* Collection Frequency:
- Total no of occurrences of a term in the collection.

* Bag of words model:

- A document is represented as a bag of words.
- Ordering of terms in a document is ignored.
- Contains no. of occurences of each term.

eg "Mary is quicker than John" is identical to "John is quicker than Mary".

* Inverse document frequency:
- Used to scale document frequency.

$$idf_t = \log \frac{N}{df_t}$$

where N = Total no. of documents in a collection.

- idf of rare term will be high
- idf of frequent term will be low.

Date

* Weighting Scheme :

— Combination of term frequency & inverse document frequency to produce a composite weight for each term in each document.

$$\text{tf-idf}_{t,d} = tf_{t,d} * idf_t$$

— Assigns a weight to a term t in document d.

① highest → t occurs many times within a small no. of documents.

② lower → t occurs fewer times in a document or occurs in many documents.

③ lowest → t occurs virtually in all documents.

$$Score(q,d) = \sum_{t \in q} \text{tf-idf}_{t,d}$$

NOTE : idf of term is always finite.

b/c $df_{t,d} \geq 1$

$idf \leq \frac{\log N}{df_{t,d}}$ → never becomes infinity.

always greater than 1

Date

* Vector Space Model.

- The representation of a set of documents as vectors in a common vector space is known as the vector space model.

- Used for IR operations including scoring documents on a query, document classification and document clustering.

→ Dot product :-

- $\vec{V}(d)$ ⇒ It is a vector derived from document d, with one component for each dictionary term

- Set of documents in a collection then viewed as a set of vectors in vector space, having one axis, for each term.

- It loses the relative ordering of terms in each document.

- Similarity between two documents is calculated using Cosine Similarity of the vector.

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|}$$

→ dot product

euclidean lengths. ←

nice

Date

Dot product of two vector $\vec{x}$ and $\vec{y}$

$\sum_{i=1}^{M} x_i y_i$.

Let $\vec{V}(d)$ = document vector for d.

M = components for d

$\vec{V}_1(d) \ldots \vec{V}_M(d)$

Euclidean length = $\sqrt{\sum_{i=1}^{M} \vec{V}_i^2(d)}$

// For concept building only.

$sim(d_1, d_2) = \hat{V}(d_1) \cdot \hat{V}(d_2)$

$\because \hat{V}(d_i) = \frac{\vec{V}(d_i)}{|\vec{V}(d_i)|}$ ∵ unit vector

example:

	Doc 1	Doc 2	Doc 3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

Euclidean length for d_1 $\sqrt{\sum_{i=1}^{4} V_i^2(d)}$

$d_1 = \sqrt{(27)^2 + (3)^2 + (14)^2} = 30.56$

$d_2 = 46.84$

$d_3 = 41.30$.

Date

Query as a vector:

- Query can also be represented as a ~~document~~ vector similar to document.
- Only terms present in a query are non-zero vector for the query.

$$score(q,d) = \frac{\vec{V}(q) \cdot \vec{V}(d)}{|\vec{V}(q)||\vec{V}(d)|}$$

* Advantages of VSM
  1. Simple model based on Linear algebra
  2. Term weights not binary
  3. Allows partial matching
  4. Rank documents according to their relevance.

* Disadvantages of VSM
  1. Loses ordering of terms.
  2. Assumes terms are statistically independent.
  3. Substrings might results in false positive match
  4. We cannot search phrases

Date________

(Food for thoughts) Chap # 06.

① Answer pg # 7, 8

② Terms: dil, Pakistan, jan, hum, sub, Ki, aur

	d_1	d_2	d_3	df	idf	$tf_1 * idf$	$tf_2 * idf$	$tf_3 * idf$
aur	0	0	1	1	0.477	0	0	0.477
dil	2	0	1	2	0.176	0.352	0	0.176
hum	0	1	0	1	0.477	0	0.477	0
jan	2	1	1	3	0	0	0	0
Ki	0	1	0	1	0.477	0	0.477	0
Pakistan	2	1	2	3	0	0	0	0
sub	0	1	0	1	0.477	0	0.477	0

Query: dil jan Pakistan.

	tf	df	tf * idf
aur	0	1	0
dil	1	2	0.176
hum	0	1	0
jan	1	3	0
Ki	0	1	0
Pakistan	1	3	0
sub	0	1	0

nice

Date

Cosine similarity of query with document 1

$$sim(\vec{v}(q), \vec{v}(d_1)) = \frac{\vec{v}(q) \cdot \vec{v}(d_1)}{|\vec{v}(q)||\vec{v}(d_1)|}$$

$$= \frac{i + j + k}{\sqrt{1^2 + 1^2 + 1^2}} \cdot \frac{2i + 2j + 2k}{\sqrt{2^2 + 2^2 + 2^2}}$$

$$= \left(\frac{i}{\sqrt{3}} + \frac{j}{\sqrt{3}} + \frac{k}{\sqrt{3}}\right)\left(\frac{i}{\sqrt{3}} + \frac{j}{\sqrt{3}} + \frac{k}{\sqrt{3}}\right)$$

$$= 1.$$

3 $g_1 = 0.2$, $g_2 = 0.31$, $g_3 = 0.49$

if term appear in $g_1 = 0.21$

" " " $g_2 = 0.31$

" " " $g_3 = 0.49$

" " " " g_1 & $g_2 = 0.51$

" " " " g_1 & $g_3 = 0.69$

" " g_2 & $g_3 = 0.8$

nice

Date

(4) $Idf = \log\left(\frac{N}{df}\right)$

$Idf = \log\left(\frac{1}{1}\right)$

$IdF = 0$

Idf will be 0 if term appears in all documents.

(5) Because $df_{t,d} \geq 1$

$idf \leq \log \frac{N}{df_{t,d}}$ → yeh kbhi b zero nhi hoga islye finite.