

Text Technologies

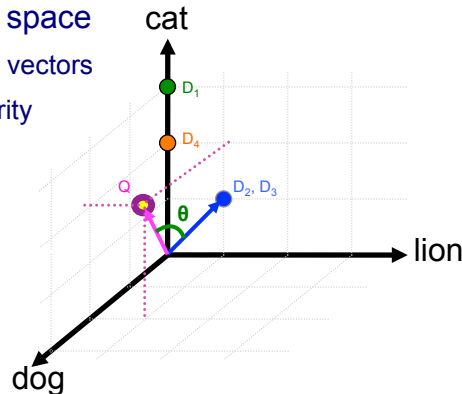
Vector Space Model of IR

Victor Lavrenko

Copyright © Victor Lavrenko, 2011

Basic VS: dimensions = words

- Separate dimension for each distinct word
 - words = coordinate vectors
 - value along dimension "cat" ~ number of times "cat" occurs
 - Comparing documents to queries
 - distance between points in space
 - Euclidian, or angle between vectors
 - usually expressed as similarity
- D_1 : "cat cat cat" $\rightarrow (0,3,0)$
 • D_2 : "cat lion" $\rightarrow (0,1,1)$
 • D_3 : "lion cat" $\rightarrow (0,1,1)$
 • D_4 : "cat cat" $\rightarrow (0,2,0)$
 • Q : cat cat lion dog dog
 $\rightarrow (2,2,1)$



Copyright © Victor Lavrenko, 2011

Vector Space Model

- Everything is a vector in some high-dimensional space
 - words, documents, queries, user preferences
- Issues to consider
 - what are the dimensions of that space (basis vectors)?
 - how to project words/documents/queries to that space?
 - how to compare documents and queries?
- Dimensions
 - basis: set of linearly-independent (orthogonal) vectors
 - "core" semantic concepts: works on toy datasets
 - words in the corpus: one dimension per distinct word
 - not orthogonal, huge dimensionality, constantly-growing

Copyright © Victor Lavrenko, 2011

Vector similarity measures

$s(Q,D)$

set-based

(word present or absent)

weighted vectors

(word has a weight in the document / query)

- inner product

$$|Q \cap D|$$

$$\sum_w Q_w D_w$$

- Jacquard coefficient

$$\frac{|Q \cap D|}{|Q \cup D|}$$

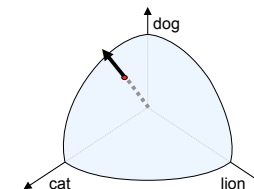
$$\frac{\sum_w Q_w D_w}{\sum_w Q_w^2 + \sum_w D_w^2 - \sum_w Q_w D_w}$$

- Cosine coefficient

$$\frac{|Q \cap D|}{\sqrt{|Q|} \cdot \sqrt{|D|}}$$

$$\frac{\sum_w Q_w D_w}{\sqrt{\sum_w Q_w^2} \cdot \sqrt{\sum_w D_w^2}}$$

- differences minor compared to how you set Q_w, D_w



Normalize to unit length \rightarrow all rank-equivalent to dot-product

Copyright © Victor Lavrenko, 2011

Term Weighting

- Term weight: relative importance of term in a doc
 - D_w : coordinate of D along dimension w $s(Q,D) = \sum_w Q_w \cdot D_w$
- Observation 1: presence / absence most important
 - weight=1 if word present, 0 otherwise $s(Q,D) = \sum_w 1_{w \in Q} \cdot 1_{w \in D}$
 - document = binary vector = set (word overlap / coordination level)
- Observation 2: key words tend to be repeated in a doc
 - $tf_{w,D}$ = number of times w occurred in D $s(Q,D) = \sum_w tf_{w,Q} \cdot tf_{w,D}$
- Observation 3: biased towards long documents
 - long docs \rightarrow higher tf , spurious word occurrences
 - normalize by document length $|D|$ $s(Q,D) = \sum_w tf_{w,Q} \cdot \frac{tf_{w,D}}{|D|}$

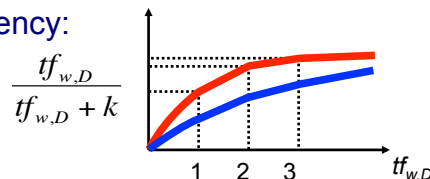
Copyright © Victor Lavrenko, 2011

Frequency Normalization

- Observation 5:
 - $D_1 = \{cryogenic, labs\}$, $D_2 = \{cryogenic, cryogenic\}$
 - $Q = \{cryogenic, labs\}$... which document is more relevant?
 - but which one is ranked higher? ($df_{labs} > df_{cryogenic}$)

- Correction:
 - first occurrence more important than a repeat (why?)
 - “squash” the growth of term frequency:

- large $K \rightarrow$ line (no squash)
- small $K \rightarrow$ step function



- Observation 6:
 - repetitions important in long docs
 - make it reflect document length $\frac{tf_{w,D}}{tf_{w,D} + \frac{k|D|}{avg|D|}}$
 - tune $k!$

Copyright © Victor Lavrenko, 2011

Inverse Document Frequency

- Observation 4: rare words carry more meaning
 - *cryogenic, apollo, jacquard* ... topical content
 - *said, went, of, the, big* ... linguistic glue
 - give more weight to rare words: $\log \frac{|C|}{df_w}$
 - $|C|$... number of documents in collection
 - df_w ... number of documents containing w
- new similarity formula: $s(Q,D) = \sum_w \underbrace{tf_{w,Q}}_{Q_w} \cdot \underbrace{\frac{tf_{w,D}}{|D|}}_{D_w} \cdot \log \frac{|C|}{df_w}$
 - tf component
 - idf component
- Inverse Document Frequency (*idf*)
 - very effective heuristic for picking out important words
 - sometimes *idf* used on the query weights
 - logarithm: to put *idf* on the same scale as the *tf* component

Copyright © Victor Lavrenko, 2011

tf.idf weighted sum

If word is repeated in the query, it's probably important

Repetitions of query words in the document \rightarrow good

Rare words more important

$$s(Q,D) = \sum_w \underbrace{tf_{w,Q}}_{\text{The more query words we match, the better}} \cdot \underbrace{\frac{tf_{w,D}}{tf_{w,D} + \frac{k|D|}{avg|D|}}}_{\text{Repetitions of same word less important than different words. Except in very long documents}} \cdot \underbrace{\log \frac{|C|}{df_w}}_{\text{Rare words more important}}$$

- Rank documents in order of decreasing $s(D,Q)$
- State-of-the-art ranking formula for short queries
 - variations actively used by many search engines
 - for long queries or doc-to-doc similarities use *tf.idf* weighted cosine

Copyright © Victor Lavrenko, 2011

Example: weighted cosine

- $D = 0.5 * \text{cat} + 0.8 * \text{dog} + 0.3 * \text{lion}$

- $Q = 0 * \text{lion} + 1.5 * \text{cat} + 0.1 * \text{dog}$ $Q_w = \frac{tf_{w,Q}}{tf_{w,Q} + k|Q|/avg.df} \cdot \log \frac{|C|}{df_w}$

- Cosine coefficient:

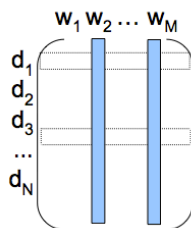
$$s(D, Q) = \frac{\sum_w D_w \cdot Q_w}{\sqrt{\sum_w D_w^2 \cdot \sum_w Q_w^2}} = \frac{0.5 \cdot 1.5 + 0.8 \cdot 0.1}{\sqrt{(0.5^2 + 0.8^2 + 0.3^2) \cdot (1.5^2 + 1^2)}} = \frac{1.55}{\sqrt{0.98 \cdot 3.25}} = 0.868$$

- number between 0 and 1 (cosine of the angle between Q,D)

- 1 iff vectors are identical (exact match between Q and D)
 - probably relevant to the query
- 0 iff vectors are orthogonal (no match)
- no negative values: vectors constrained to positive weights

Uses of VSM

- Not just ranking documents in response to query
- Any time you want to know if text A is similar to text B:
 - does this essay look like the writing of author B?
 - does patent A infringe on any part of patent portfolio B?
 - does email A look more like spam emails?
 - is this piece of code similar to any part of system B?
- Determine if word/phrase A is similar to word/phrase B:
 - inverted list for a word is a vector over documents
 - similarity("cat", "lion") = cosine of their inverted lists
 - need to customize weighting (no idf, length=df, etc.)

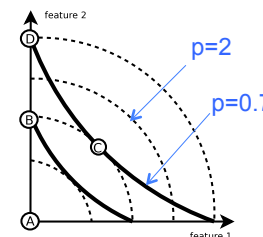


More distance / similarity measures

- p-norm distance:

- p=2: Euclidian
- p=1: Manhattan
- p=0: $\max |Q_w - D_w|$ = logical OR
- p=∞: $\min |Q_w - D_w|$ = logical AND

$$\sqrt[p]{\sum_w |Q_w - D_w|^p}$$



- Treat documents / queries as word histograms

- Q_w, D_w non-negative, add up to 1

- KL divergence: $-\sum_w Q_w \log \frac{Q_w}{D_w}$ $\chi^2: \frac{1}{2} \sum_w \frac{|Q_w - D_w|^2}{|Q_w + D_w|}$

- <http://bit.ly/r6R0RY>

- Remember to convert distance to similarity (e.g. negate)
- Still need to set weights (Q_w, D_w) appropriately!

Summary

- Everything is a vector, one dimension per word
- Rank by similarity of document vectors to query
 - dot product or cosine of the angle between vectors
- Term weighting: very important, *tf.idf* is universal
- Heuristic in nature:
 - easy to assimilate good ideas from other retrieval models
 - components not interpretable → no guide what to try next
 - encourages ad-hoc engineering: tweak, test out, tweak ...
 - no notion of relevance (= similarity?)
- Very popular, hard to beat