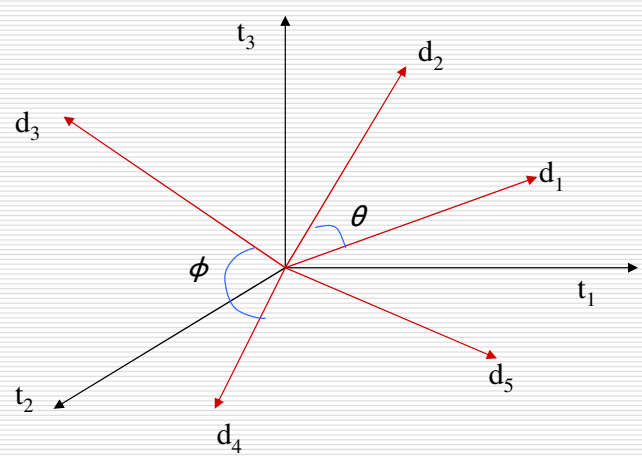
The Vector Space Model
Salton 1965

- Document d_i and query q_j are viewed as vectors in a n-dimensional space, where n is the dictionary size

- Why turn docs into vectors?

  First application: Query-by-example
  - Given a doc D, find others "like" it.
  - Now that D is a vector, find vectors (docs) "near" it.

Intuition



Postulate: Documents that are "close together" in the vector space talk about the same things.

The vector space model

Query as vector:

- We regard query as short document
- We return the documents ranked by the closeness of their vectors to the query, also represented as a vector.

Desiderata for proximity

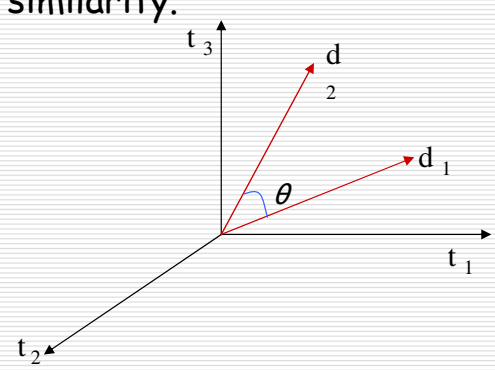
- If d_1 is near d_2, then d_2 is near d_1.
- If d_1 near d_2, and d_2 near d_3, then d_1 is not far from d_3.
- No doc is closer to d than d itself.

First cut

- Distance between d_1 and d_2 is the length of the vector $|d_1 - d_2|$.
  - Euclidean distance
- Why is this not a great idea?
- We still haven't dealt with the issue of length normalization
  - Long documents would be more similar to each other by virtue of length, not topic
- However, we can implicitly normalize by looking at *angles* instead

Cosine similarity

- Distance between vectors d_1 and d_2 *captured* by the cosine of the angle x between them.
- Note – this is *similarity*, not distance
  - No triangle inequality for similarity.



Cosine similarity

- A vector can be *normalized* (given a length of 1) by dividing each of its components by its length – here we use the L_2 norm $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$

$$\left|\vec{d}_j\right| = \sqrt{\sum_{i=1}^{n} w_{i,j}} = 1$$

- This maps vectors onto the unit sphere: longer documents don't get more weight