The Vector Space Model

Documents as vectors

- we have |V|-dimensional vector space
    - V 는 사전(vocab)에 들어있는 단어 수
- **Terms are axes of the space**
- Documents are points or vectors in this space
- **very high-dimensional** : tens of millions of dimensions when you apply this to a web search engine
- These are very **sparse vectors** - most entries are **zero**

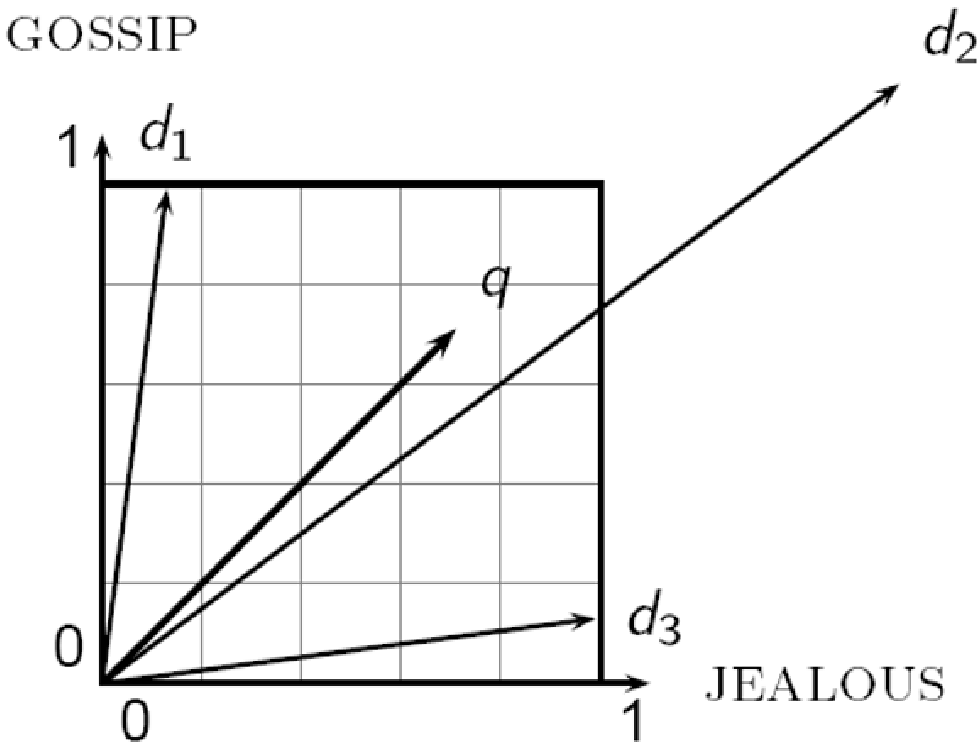
Queries as vectors

- **Key idea 1** : Do the same for queries : represent them as vectors in the space
- **Key idea 2** : Rank documents according to their proximity to the query in this space

- proximity = similarity of vectors
- proximity $\approx$ inverse of distance
- Recall : We do this because we want to **get away from** the you're either in or out **Boolean Model**
- Instead : rank more relevant documents higher than less relevant documents

Formalizing vector space proximity

- First cut : distance between two points
- Euclidean distance?
    - Euclidean distance is bad idea... $\leftarrow$ because Euclidean distance is large for vectors of different lengths
    - Example : q는 d_2와 가장 유사함에도 불구하고 Euclidean distance로 측정할 때 가장 거리가 크다.



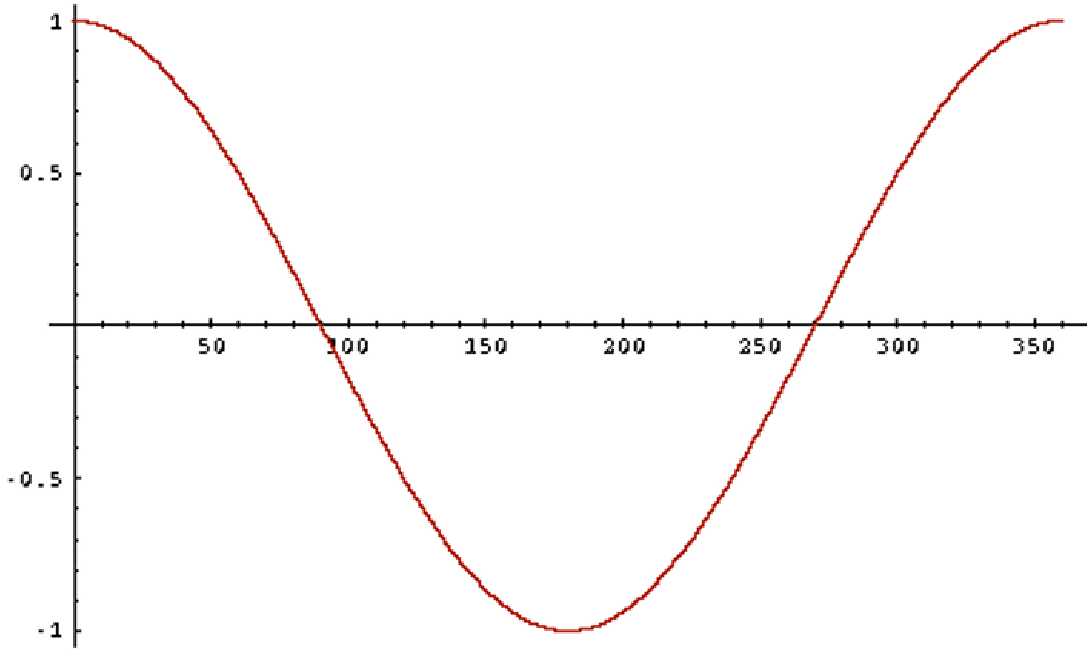
$\rightarrow$ start looking at the **angle** in the vector space

Use angle instead of distance

- Thought experiment : take a document d and append it to itself. Call this document d'
- "Semantically" d and d' have the same content : 즉, d에 비해 d'의 크기가 2배 클 뿐 (벡터의 길이가 2배 길어짐) 의미상의 내용은 일치히다.
- The Euclidean distance between the two documents can be quite large
- **The angle between the two documents is 0**, corresponding to maximal similarity.
- **Key idea : rank documents according to angle with query**

From angles to cosines

- The following **two notions are equivalent.**
  - Rank documents in decreasing order of the angle between query and document
  - Rank documents in increasing order of cosine(query, document)
- Cosine is a monotonically decreasing function for the interval $[0°, 180°]$ $\rightarrow$ 1에서 -1의 값을 갖는다. angle의 차이가 0일때는 1의 값을, angle의 차이가 180일때는 -1의 값을 가진다



Length Normalization

- A vector can be normalized by dividing each of its components by its length - for this we use the L_2 norm :

$$||\vec{x}||_2 = \sqrt{\sum_i x_i^2}$$

- **Dividing a vector by its L_2 norm makes it a unit (length) vector** (on surface of unit hypersphere)
    - unit vector : a vector of length 1

Cosine (query, document) $\rightarrow$ Length Normalization

Dot product   Unit vectors

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q} \bullet \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|} \bullet \frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

- q_i is the tf-idf weight of term i in the query
- d_i is the tf-idf weight of term i in the document
- $\cos(\vec{q},\vec{d})$ is **the cosine similarity of $\vec{q}$ and $\vec{d}$** or **the cosine of the angle between $\vec{q}$ and $\vec{d}$**

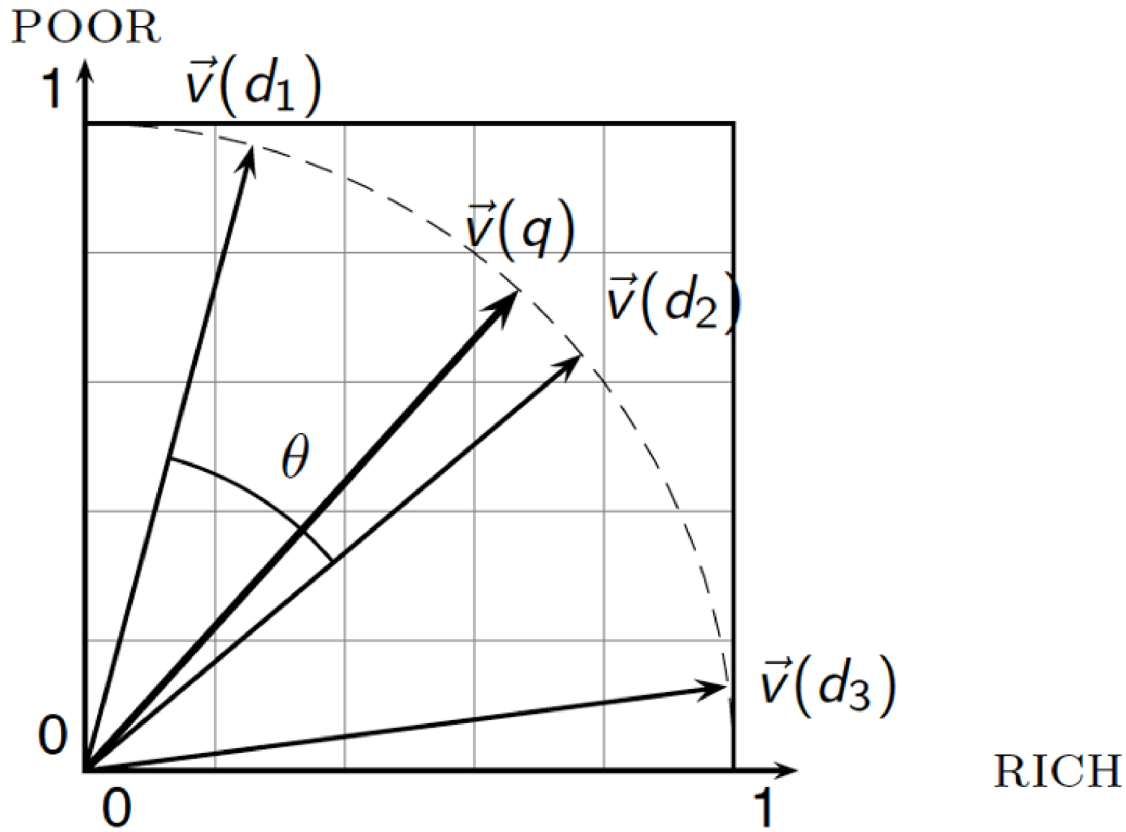
Cosine for length-normalized vectors

- For length-normalized vectors, cosine similarity is simply the dot product (or scalar product) :

$$\cos(\vec{q},\vec{d}) = \vec{q} \bullet \vec{d} = \sum_{i=1}^{|V|} q_i d_i$$

for q, d length-normalized.

Cosine Similarity illustrated



- cosine similarity를 이용한 q와 가장 유사한 document는 d_2

Cosine Similarity amongst 3 documents

- How similar are the novels
  - SaS : Sense and Sensibility
  - PaP : Prede and Prejudice
  - WH : Wuthering Heights?

term	SaS	PaP	WH
affection	115	58	20
jealous	10	7	11
gossip	2	0	6
wuthering	0	0	38

Term frequencies (counts)

Log frequency weighting

term	SaS	PaP	WH
affection	3.06	2.76	2.30
jealous	2.00	1.85	2.04
gossip	1.30	0	1.78
wuthering	0	0	2.58

After length normalization

term	SaS	PaP	WH
affection	0.789	0.832	0.524
jealous	0.515	0.555	0.465
gossip	0.335	0	0.405
wuthering	0	0	0.588

cos(SaS,PaP) ≈
$0.789 \times 0.832 + 0.515 \times 0.555 + 0.335 \times 0.0 + 0.0 \times 0.0$
≈ 0.94

cos(SaS,WH) ≈ 0.79

cos(PaP,WH) ≈ 0.69

Why do we have cos(SaS,PaP) > cos(SaS,WH)?