# BUS 243

Lecture 2: Vector Space Models

# RANKED RETRIEVAL

- Consider the problem of Boolean search in the case of large corpus
  - The number of matching documents could be too large
- Ranked retrieval: order documents by how likely they are to be relevant to the information need
  - Estimate relevance score of query, document pair $(q, d_i)$
  - Sort documents by relevance
  - Display sorted results
- How do we estimate relevance?

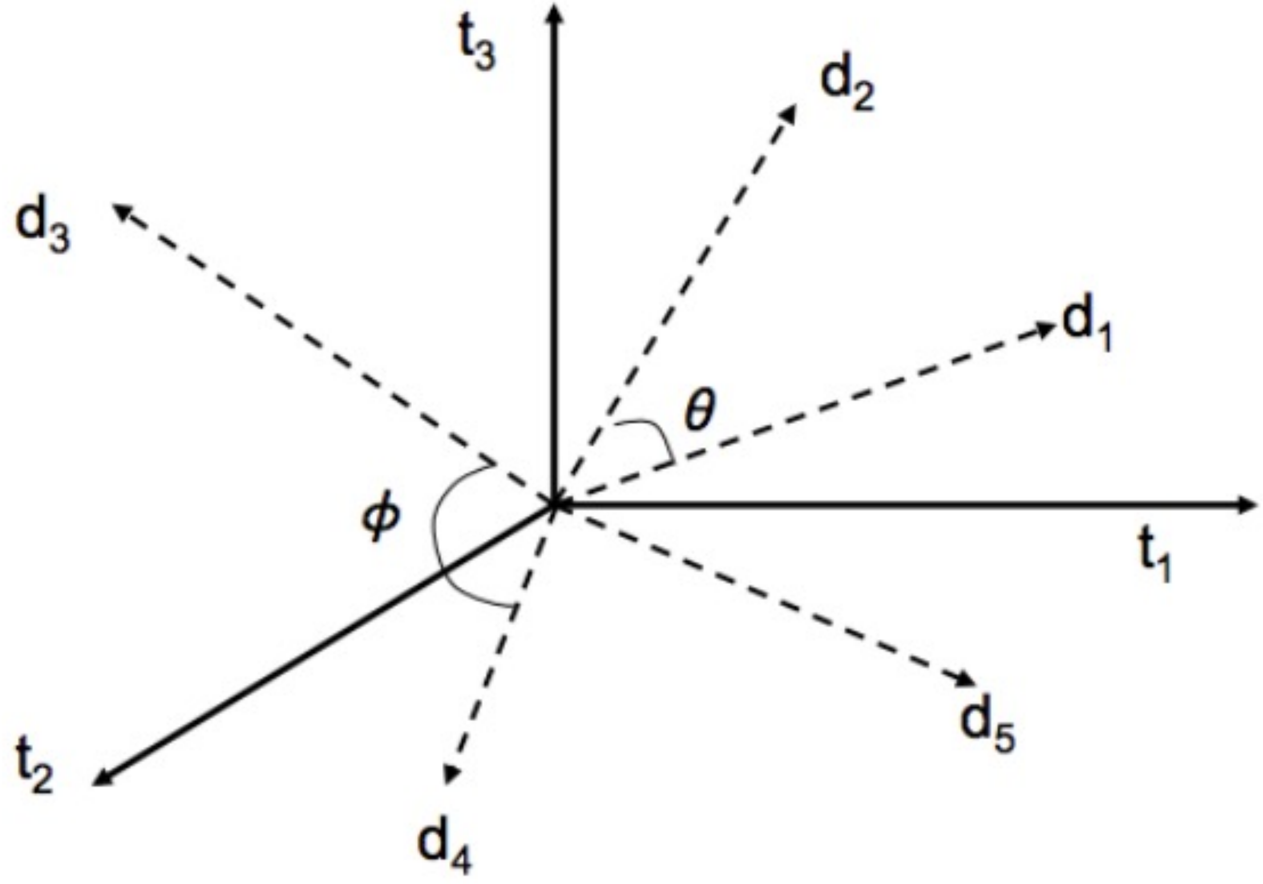
- Assume document is relevant if it has a lot of query terms
  - Obviously, it is too strong assumption. Why?
    - Structure
    - The ordering of the terms in a document is ignored
- Replace relevance with $sim(q, d_i)$
- Compute similarity of vector representations

# TEXT REPRESENTATION

- Central problem in NLP is how to store /represent / query text
- This is almost certainly wrong model… hopefully useful
- Modern NLP typically uses vector representations

# TERM FREQUENCY

- Consider how to represent a document
- One way is to assign each term in a document a weight
- Thus far, view a document as a sequence of terms
  - assign a weight equal to the number of occurrences of term t in d
  - called term frequency, $\mathrm{tf}_{t,d}$
- In this view of a document, known in the literature as *the bag of words model*



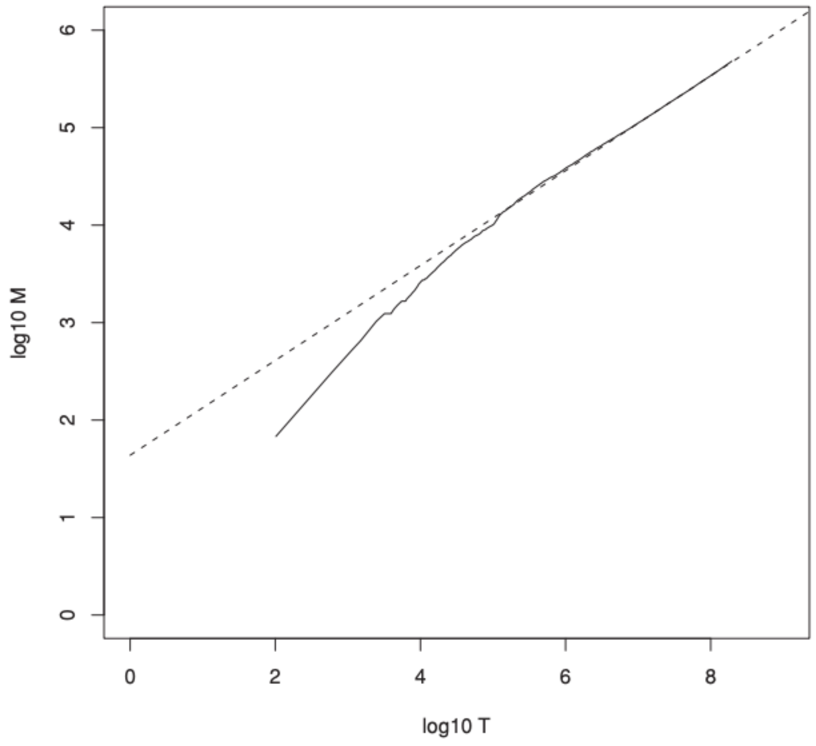

- Each document is a vector $d_i = (w_{i,1}, \ldots, w_{i,V})$

# STATISTICAL PROPERTIES OF TERMS IN IR

- The number of terms is the main factor in determining the size of the dictionary
- Compressing the dictionary is of interest, but need to estimate:
  - The number of distinct terms $M$ in a corpus
  - Oxford English Dictionary defines more than 600,000 words, but the vocab of most large corpus is much large… Why?

# HEAPS' LAW

- Estimate vocab size:
  - $M = kT^b$
    - $M$: The number of distinct terms in a corpus
    - $T$: the number of tokens in a corpus
    - $k$ and $b$: parameters
- Reuters Corpus Volume I (RCV1)
  - $b$=0.49 and $k$ = 44
  - For the first 1,000,020 tokens, Heaps' predicts 38,323 terms
  - Actual number is 38,365



- So what's the problem of bag of words model?
- We may need to implement a mechanism to reduce the impact of frequently occurring terms in the collection
  - Document frequency $\mathrm{df}_t$: the number of documents in the collection including a term t
- Define the inverse document frequency (idf) of a term t
  - $\mathrm{idf}_t = \log \frac{N}{\mathrm{df}_t}$
- How is this used to scale the term weight?

- Let’s say, you have a corpus of 1 million documents
  - Consider the term “cat”
  - Suppose you have exactly 1 document that contains it
    - The raw IDF: 1,000,000/1
  - Now suppose there are 10 documents with the term “dog”
    - The raw IDF: 1,000,000/10 = 100,000
- The base of log function is not important, because you only want to make the frequency distribution uniform, not to scale it within a particular numerical range

# INTUITIONS

- Term weights consist of two components
  - Local: how important is the term in this document?
  - Global: how important is the term in the collection?
- Here's the intuition:
  - Terms that appear often in a document should get high weights
  - Terms that appear in many documents should get low weights
- How do we capture this mathematically?
  - Term frequency (local)
  - Inverse document frequency (global)

# TF-IDF TERM WEIGHTING

- $w_{t,d} = \mathrm{tf}_{t,d} \times \log \frac{N}{\mathrm{df_t}}$
  - Term *t*'s weight in document *d*
  - Frequency of word *t* in document *d*
  - Total number of document *N*
  - Number of documents *t* appears in

# FREQUENCY OF TERMS (ZIPF'S LAW)

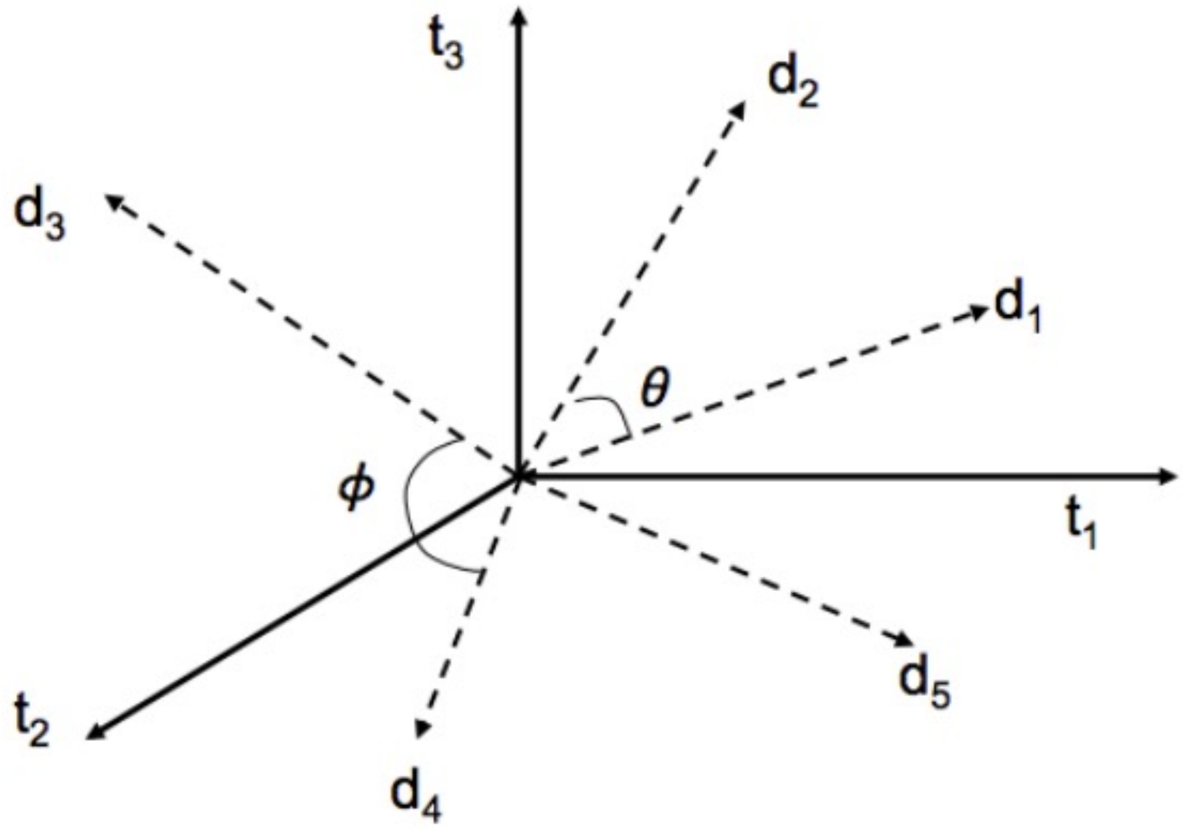
- The most frequent words ("the") are everywhere but useless for queries.
- The most useful words are relatively rare . . . but there are lots of them
  - $f_t = \frac{c}{R_t}$
    - The frequency of a term t is inversely proportional to
    - The rank (in frequency) of term
    - Scaled by a constant
  - Can't just throw out useless words

# VECTOR REASONING

- Introduce the overlap score measure
  - $\text{Score}(q, d) = \sum_{t \in q} \text{tfidf}_{t,d}$
  - $q$: query
  - $t$: term
  - $d$: document
- What's intuition?

# VECTOR SPACE MODEL FOR SCORING

- Denote $\vec{V}(d)$ the vector derived from document
  - Each component is the weight for each vocabulary term
  - Vector space model
    - Salton 1975
- The set of documents in a corpus may be viewed as a set of vectors in a vector space, in which there is one axis for each term




- Now consider the angle between vectors

# SIMILARITY METRIC

- Angle between vectors
  - $\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|} = \vec{v}(d_1) \cdot \vec{v}(d_2)$

# EXAMPLE

- Corpus
  - Doc 0: The sky is blue
  - Doc 1: The sun is bright today
  - Doc 2: The sun in the sky is bright
  - Doc 3: We can see the shining sun the bright sun

## How many docs did each term appear in?

| Doc Frequency | |
| --- | --- |
| blue | 1.00 |
| bright | 3.00 |
| can | 1.00 |
| **in** | 1.00 |
| **is** | 3.00 |
| see | 1.00 |
| shining | 1.00 |
| sky | 2.00 |
| sun | 3.00 |
| the | 4.00 |
| today | 1.00 |
| we | 1.00 |

# TERM FREQUENCY

- Original Salton paper uses absolute frequency and makes vectors unit length later
  - let's use raw frequency immediately.

| | | | | |
|---|---|---|---|---|
| blue | 0.25 | 0.00 | 0.00 | 0.00 |
| bright | 0.00 | 0.20 | 0.14 | 0.11 |
| can | 0.00 | 0.00 | 0.00 | 0.11 |
| **in** | 0.00 | 0.00 | 0.14 | 0.00 |
| **is** | 0.25 | 0.20 | 0.14 | 0.00 |
| see | 0.00 | 0.00 | 0.00 | 0.11 |
| shining | 0.00 | 0.00 | 0.00 | 0.11 |
| sky | 0.25 | 0.00 | 0.14 | 0.00 |
| sun | 0.00 | 0.20 | 0.14 | 0.22 |
| the | 0.25 | 0.20 | 0.29 | 0.22 |
| today | 0.00 | 0.20 | 0.00 | 0.00 |
| we | 0.00 | 0.00 | 0.00 | 0.11 |

# TF-IDF

- Use log base 10

| | | | | |
|---|---|---|---|---|
| bright | 0.00 | 0.02 | 0.02 | 0.01 |
| sun | 0.00 | 0.02 | 0.02 | 0.03 |
| today | 0.00 | 0.12 | 0.00 | 0.00 |
| can | 0.00 | 0.00 | 0.00 | 0.07 |
| **is** | 0.03 | 0.02 | 0.02 | 0.00 |
| blue | 0.15 | 0.00 | 0.00 | 0.00 |
| sky | 0.08 | 0.00 | 0.04 | 0.00 |
| **in** | 0.00 | 0.00 | 0.09 | 0.00 |
| we | 0.00 | 0.00 | 0.00 | 0.07 |
| the | 0.00 | 0.00 | 0.00 | 0.00 |
| see | 0.00 | 0.00 | 0.00 | 0.07 |
| shining | 0.00 | 0.00 | 0.00 | 0.07 |

# QUERY DOCUMENT

- The shining sky ball
- Don’t use UNK (unknown) token
- Query
  - `the`: 0.0
  - `shining`: 0.2
  - `sky`: 0.1
  - ?

- Term frequencies
  - $\mathrm{tf}_{the} = 0.33$
  - $\mathrm{tf}_{shining} = 0.33$
  - $\mathrm{tf}_{sky} = 0.33$
- Document frequencies
  - $\mathrm{df}_{the} = 4$
  - $\mathrm{df}_{shining} = 1$
  - $\mathrm{df}_{sky} = 2$

- $\mathrm{tfidf}_{the} = \frac{1}{3}\log_{10}\frac{4}{4} = 0$
- $\mathrm{tfidf}_{shining} = \frac{1}{3}\log_{10}\frac{4}{1} = 0.200486$
- $\mathrm{tfidf}_{sky} = \frac{1}{3}\log_{10}\frac{4}{2} = 0.100243$

# MOST SIMILAR DOCUMENT?

- Score(q,d)= $\sum_{t \in q} \mathrm{tfidf}_{t,d}$
  - Doc 0: The sky is blue → 0.008
  - Doc 1: The sun is bright today → 0.0
  - Doc 2: The sun in the sky is bright → 0.004
  - Doc 3: We can see the shining sun the bright sun → 0.013