Natural Language Processing (NLP)

Vector Space Model (VSM)

- Generalization of the Bag of Words model
- Each document from the corpus is represented as a multi- dimensional vector
  - Each unique term from the corpus represents one dimension of the vector space
  - *Term* can be a single word or a sequence of words (phrase)
  - The number of unique terms in the corpus determines the dimension of the vector space

Vector Space Model (VSM)

- Vector elements are weights associated with individual terms;
- weights reflect the relevancy of the corresponding terms in the given corpus

Vector Space Model (VSM)

- If a corpus consists of *n* terms (t_i, $i=1,n$), document *d* from that corpus would be represented with the vector: $d = \{w1,w2,\ldots,wn\}$, where *wi* are weights associated with terms *ti*

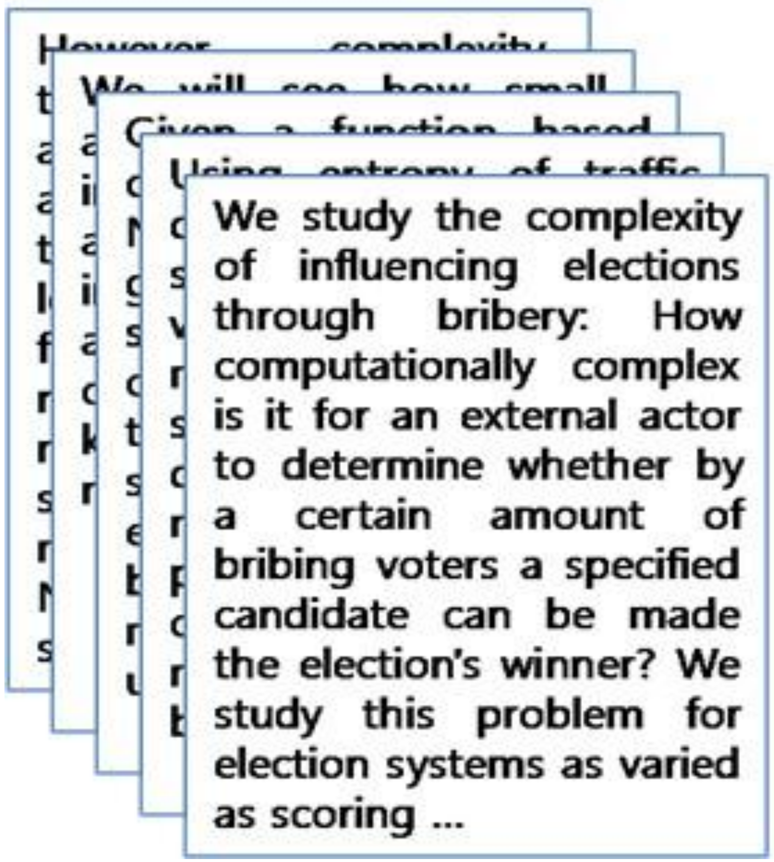
Vector Space Model (VSM)

In VSM, corpus is represented in the form of ***Term Document Matrix*** *(TDM)*, i.e., an *m* x *n* matrix with following features:

- Rows (*i=1,m*) represent terms from the corpus
- Columns (*j=1,n*) represent documents from the corpus
- Cell *ij* stores the weight of the term *i* in the context of the document *j*

Vector Space Model (VSM)

Documents



Vector-space representation

	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term-document matrix

Vector Space Model (VSM)

- Before creating the TDM matrix, documents from the corpus need to be *preprocessed*
- Rationale / objective: to reduce the set of words to those that are expected to be the *most relevant* for the given corpus

Vector Space Model (VSM)

– Preprocessing (often) includes:
  - Normalizing the text
  - Removing terms with very small / high frequency in the given corpus
  - Removing the so-called stop-words
  - Reducing words to their root form through *stemming or lemmatization*

Vector Space Model (VSM)

Normalization Of Text

– **Objective:** transform various forms of the same term into a common, ‘normalized’ form

- E.g.: Apple, apple, APPLE -> apple
- Intelligent Systems, Intelligent systems, Intelligent-systems **->** intelligent systems

Vector Space Model (VSM)

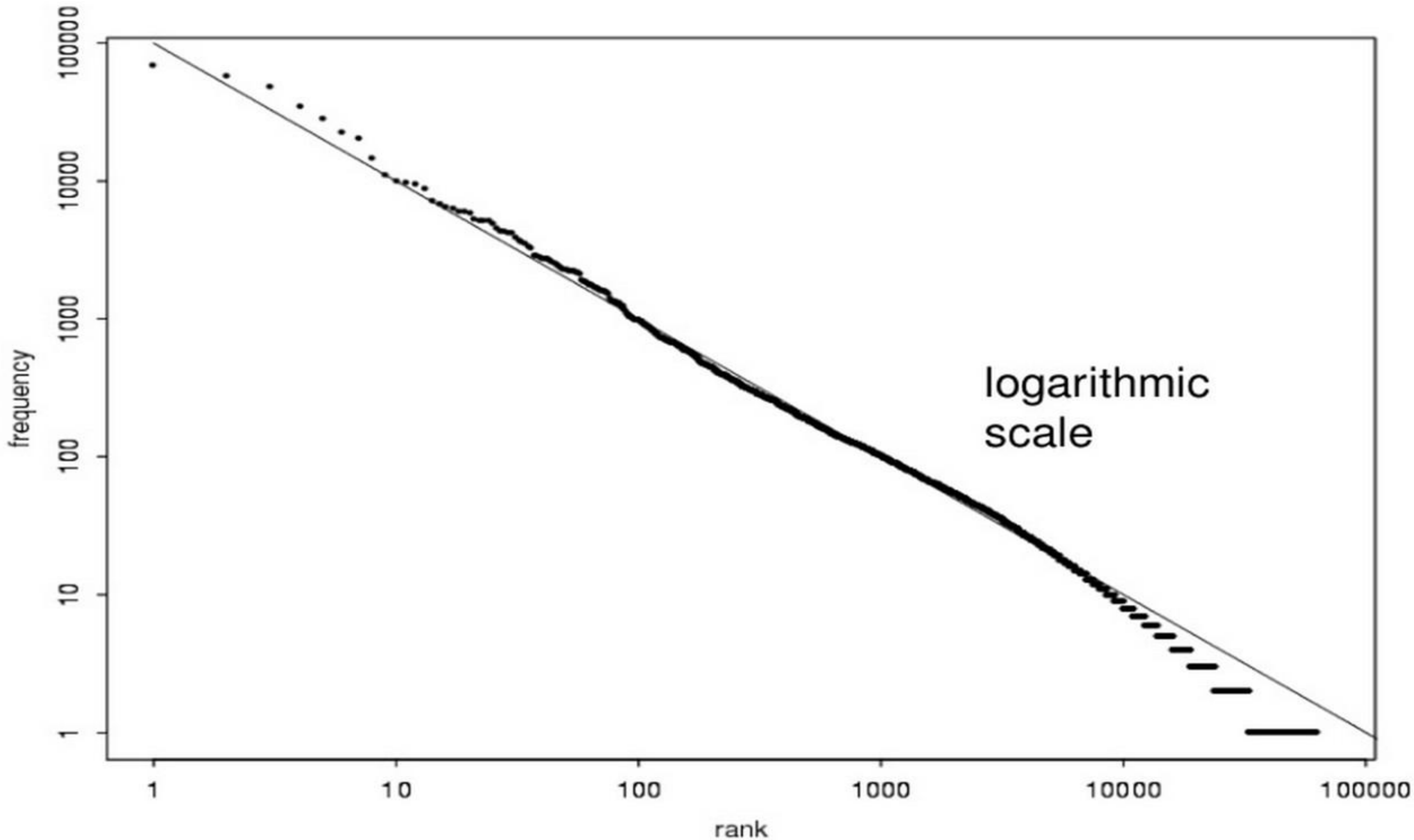
Normalization Of Text

- **How it is done:**
  - Using simple rules:
    - Remove all punctuation marks (dots, dashes, commas,…)
    - Transform all words to lower case
  - Using a dictionary, such as **WordNet,** to replace synonyms with a common, often more general, concept
    - E.g., “automobile, car” -> vehicle

Vector Space Model (VSM)

Removing High And Low Frequency Terms

- **Empirical observations (in numerous corpora):**
  - Many low frequency words
  - Only a few words with high frequency
- Formalized in the ***Zipf's rule***:
  - the frequency of a word in a given corpus is inversely proportional to its rank in the frequency table (for that corpus)


100000
10000
1000
100
10
1
frequency
1
10
100
1000
10000
100000
rank
logarithmic
scale

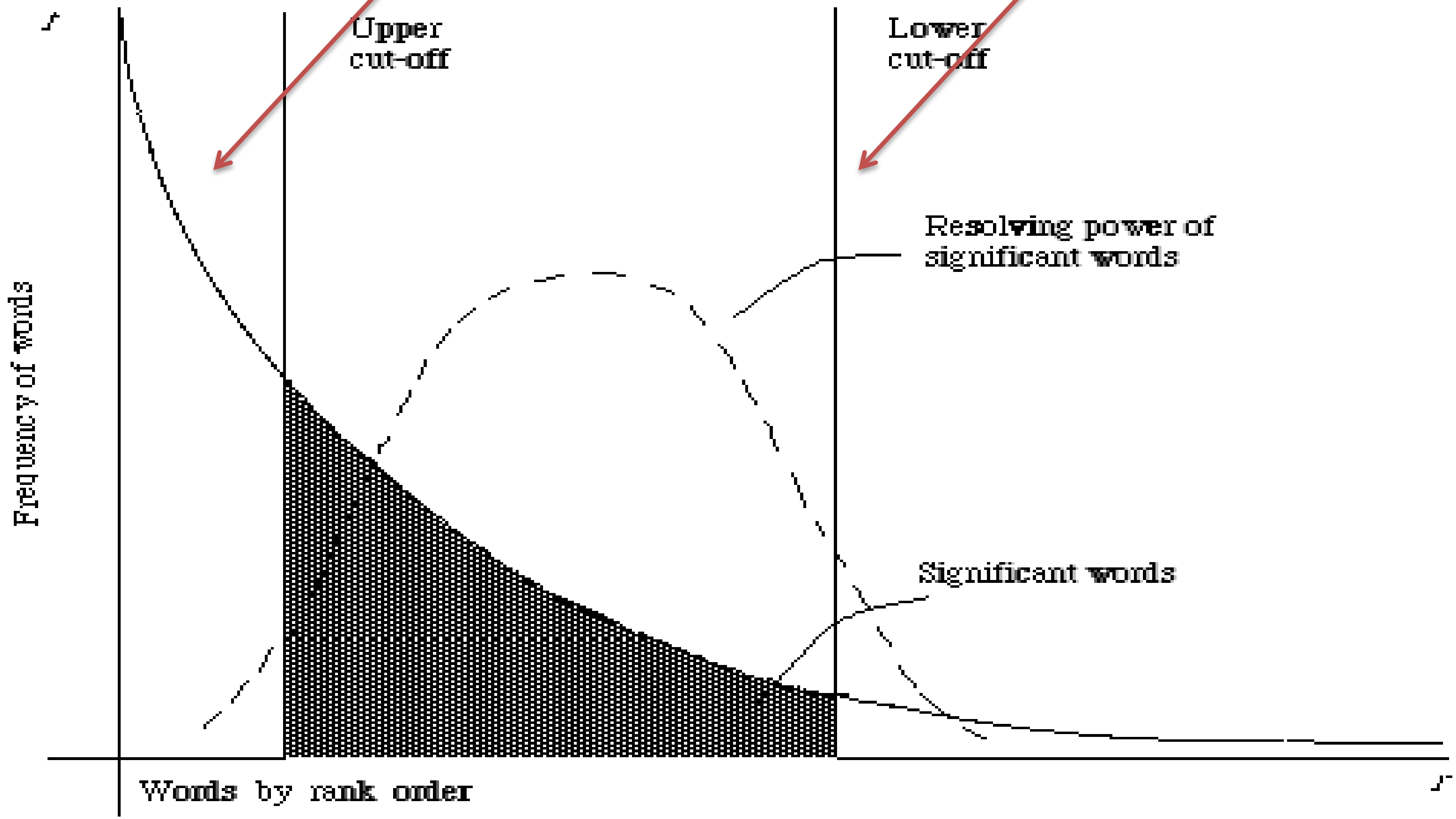
Vector Space Model (VSM)

Implications Of The Zipf's Rule

- high frequency but are semantically almost useless
    - Examples: the, a, an, we, do, to
- semantically rich, but are of very low frequency
    - Example: dextrosinistral
- The rest of the words are those that represent the corpus the **best** and thus should be **included** in the VSM model

Note : dextrosinistral=expert in physical movement with hands

Remove words that do not bear meaning



Vector Space Model (VSM)

– Stop-words are those words that (on their own) do not bear any information / meaning

– It is estimated that they represent 20-30% of words in any corpus

– There is no unique stop-words list

- Frequently used lists are available at: http://www.ranks.nl/stopwords

– Potential problems with stop-words removal:

- the loss of original meaning and structure of text
- examples: “this is not a good option” -> “option”
- “to be or not to be” -> null

Lemmatization And Stemming ?

Already discussed in tutorial

Vector Space Model (VSM)

Computing Terms' Weights

- Simple and frequently used approaches include:
  - Binary weights
  - Term Frequency (TF)
  - Inverse Document Frequency (IDF)
  - TF-IDF

Vector Space Model (VSM)

Binary Weights

- Weights take the value of 0 or 1, to reflect the presence (1) or absence (0) of the term in a particular document

Vector Space Model (VSM)

Binary Weights

	text	information	identify	mining	mined	is	useful	to	from	apple	delicious
Doc1	1	1	1	1	0	1	1	1	0	0	0
Doc2	1	1	0	0	1	1	1	0	1	0	0
Doc3	0	0	0	0	0	1	0	0	0	1	1

Example:

Doc1: Text mining is to identify useful information.
Doc2: Useful information is mined from text.
Doc3: Apple is delicious.

Vector Space Model (VSM)

Term Frequency

TF represents the frequency of the term in a specific document

- The underlying assumption: the higher the term frequency in a document, the more important it is for that document
- TF(t) = c(t,d)
- *c(t,d)* – the number of occurrences of the term *t* in the document *d*

Vector Space Model (VSM)

Inverse Document Frequency (IDF)

- Assign higher weights to unusual terms, i.e., to terms that are not so common in the corpus
- IDF is computed at the **corpus level**, and thus describes corpus as a whole, not individual documents
- It is computed in the following way:

  $IDF(t) = 1 + \log(N/\mathit{df(t)})$

  Where, N = number of documents in the corpus

  df(t) = number of documents with the term *t*

Vector Space Model (VSM)

TF–IDF

- The underlying idea: value those terms that are not so common in the corpus (relatively high IDF), but still have same reasonable level of frequency (relatively high TF)
- The **most frequently** used metric for computing term weights in a VSM
- One popular 'instantiation' of this formula:

  $$\text{TF-IDF}(t) = tf(t) * \log(N/df(t))$$

Vector Space Model (VSM)

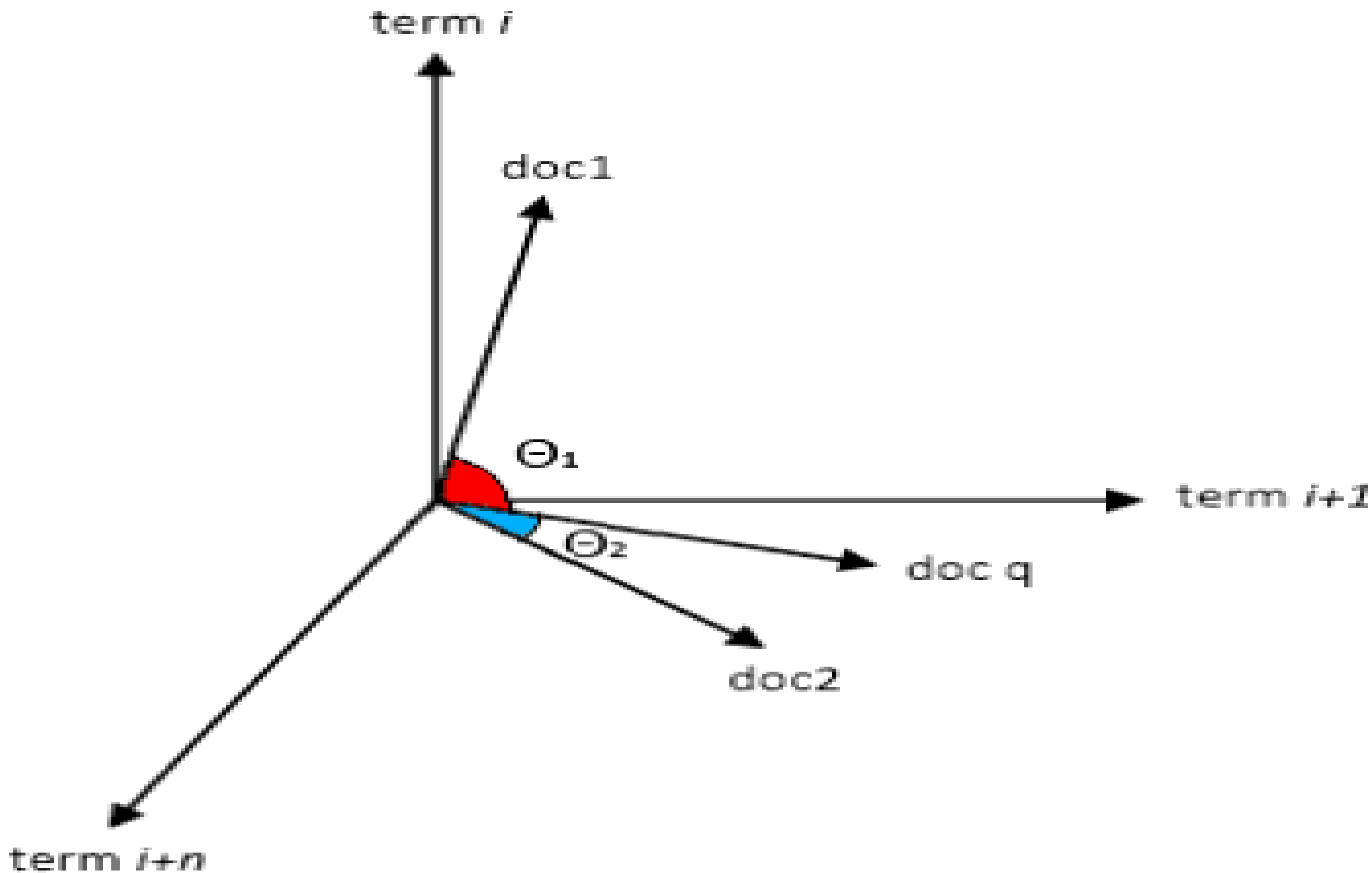
Estimating **Similarity** Of Documents

- Key question: which metric to use for estimating the similarity of documents (i.e., vectors that represent documents)?
- The most well known and widely used metric is ***Cosine similarity*** *"It is dot product of two vectors divided by the product of the two vector's lengths (or magnitudes)."*
- Cosine Similarity

  $\cos(d_i,d_j) = V_i \times V_j / (||V_i||\ ||V_j||)$

  V_i and V_j are vectors representing documents d_i and d_j

Vector Space Model (VSM)



Vector Space Model (VSM)