INTRODUCTION TO TEXT MINING

Jelena Jovanovic

Email: jeljov@gmail.com

Web: http://jelenajovanovic.net

OVERVIEW

- What is Text Mining (TM)?
- Why is TM relevant? Why do we study it?
- Application domains
- The complexity of unstructured text (the origin of TM challenges)
- *Bag-of-words* representation of text
- Vector Space Model
  - Methods/techniques for text pre-processing
  - Assessing the relevancy of individual words/phrases
  - Measuring document similarity: Cosine similarity

WHAT IS TEXT MINING (TM)?

The use of computational methods and techniques to extract high quality information from text

A computational approach to the discovery of
new, previously unknown information and/or knowledge
through automated extraction of information from
often large amounts of unstructured text

WHY IS TM RELEVANT / USEFUL ?

- Unstructured text is present in various forms, and in huge and ever increasing quantities:
  - books,
  - financial and other business reports,
  - various kinds of business and administrative documents,
  - news articles,
  - blog posts,
  - wiki,
  - messages/posts on social networking and social media sites,
  - …
- It is estimated that ~80% of all the available data are unstructured data

WHY IS TM RELEVANT / USEFUL?

- To enable effective and efficient use of such huge quantities of textual content, we need computational methods for
  - automated extraction of information from unstructured text
  - analysis and summarization of extracted information
- TM research and practice are focused on the development, continual improvement and application of such methods

TM APPLICATION DOMAINS

- Document classification*
- Clustering / organizing documents
- Document summarization
- Visualization of document space (often aimed at facilitating document search)
- Making predictions (e.g., predicting stock market prices based on the analysis of news articles and financial reports)
- Content-based recommender systems (for news articles, movies, books, articles, …)

*The term *document* refers to any kind of unstructured piece of text: blog post, news article, tweet, status update, business document, …

THE COMPLEXITY OF UNSTRUCTURED TEXT

- In general, interpretation / comprehension of unstructured content (text, images, videos) is (often) easy for people, but very complex for computer program
- In particular, difficulties with automated text comprehension are caused by the fact that the human / natural language:
  - is full of ambiguous terms and phrases
  - often strongly relies on the context and background knowledge for defining and conveying meaning
  - is full of fuzzy and probabilistic terms and phrases
  - strongly based on commonsense knowledge and reasoning
  - is influenced by and is influencing people's mutual interactions

ADDITIONAL CHALLENGES FACED BY TM

- The use of supervised machine learning (ML) methods for TM is often very expensive
  - This is caused by the need to prepare high number of annotated documents to be used as the training dataset
  - Such a training set is essential for, e.g., document classification or extraction of entities, relations and events from text
- High-dimension of the attribute space:
  - Documents are often described with numerous attributes, which further impedes the application of ML methods
  - Most often, attributes are either all terms or a selection of terms and/or phrases from the collection of documents to be analyzed

BAG OF WORDS REPRESENTATION OF TEXT

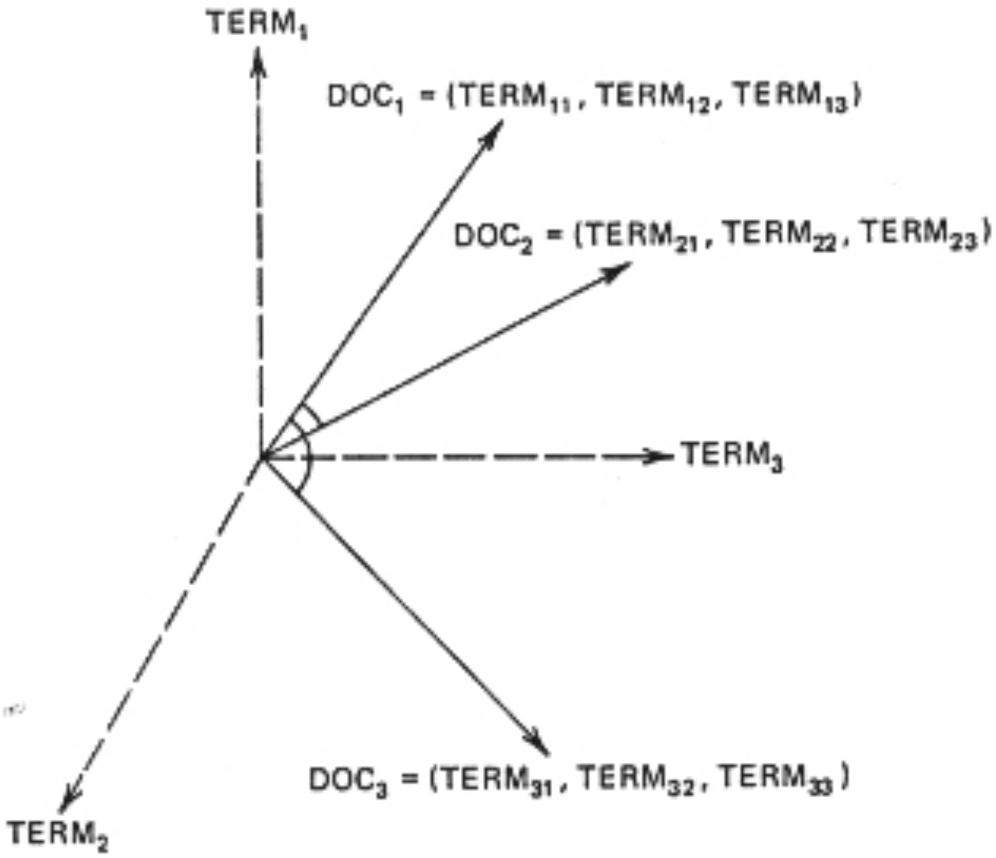
- Considers text a simple set/bag of words
- Based on the following (unrealistic) assumptions:
  - words are mutually independent,
  - word order in text is irrelevant
- Despite its unrealistic assumptions and simplicity, this approach to text modeling proved to be highly effective, and is often used in TM

BAG OF WORDS REPRESENTATION OF TEXT

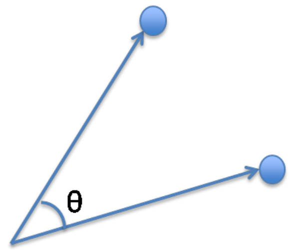
aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
…	
Zaire	0

Unique words from the corpus are used to create the corpus ‘dictionary’; then, each document from the corpus is represented as a vector of (dictionary) word frequencies

BAG OF WORDS MODEL:
COMPUTING DOCUMENTS' SIMILARITY



$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



Source:
http://www.cs.uni.edu/~okane/source/ISR/isr.html

VECTOR SPACE MODEL

- 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

**corpus* refers to a collection of documents to be processed / analyzed

VECTOR SPACE MODEL

- Vector elements are weights associated with individual terms; these weights reflect the relevancy of the corresponding terms in the given corpus
- If a corpus consists of *n* terms (t_i, $i=1,n$), document *d* from that corpus would be represented with the vector: $d = \{w_1, w_2, \ldots, w_n\}$, where w_i are weights associated with terms t_i

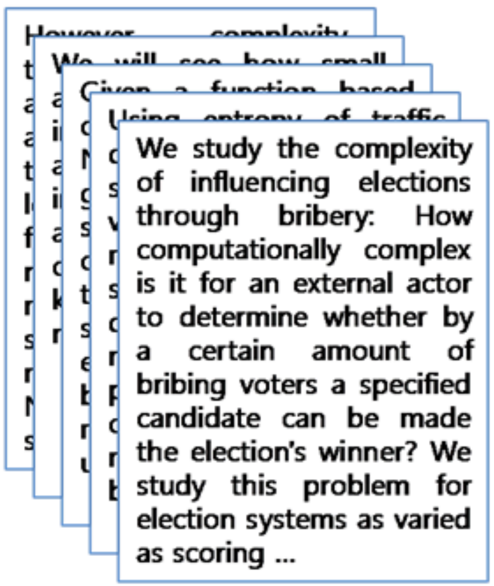
VECTOR SPACE MODEL

- Distances among vectors in this multi-dim. space represent the relationships among the corresponding documents
- It is assumed that documents that are ‘close’ to one another in this multi-dim. space, are also ‘close’ (similar) in meaning

VSM: Term Document Matrix

- 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*

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

Image source: http://mlg.postech.ac.kr/research/nmf

VSM: TEXT PREPROCESSING

- 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
- 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

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

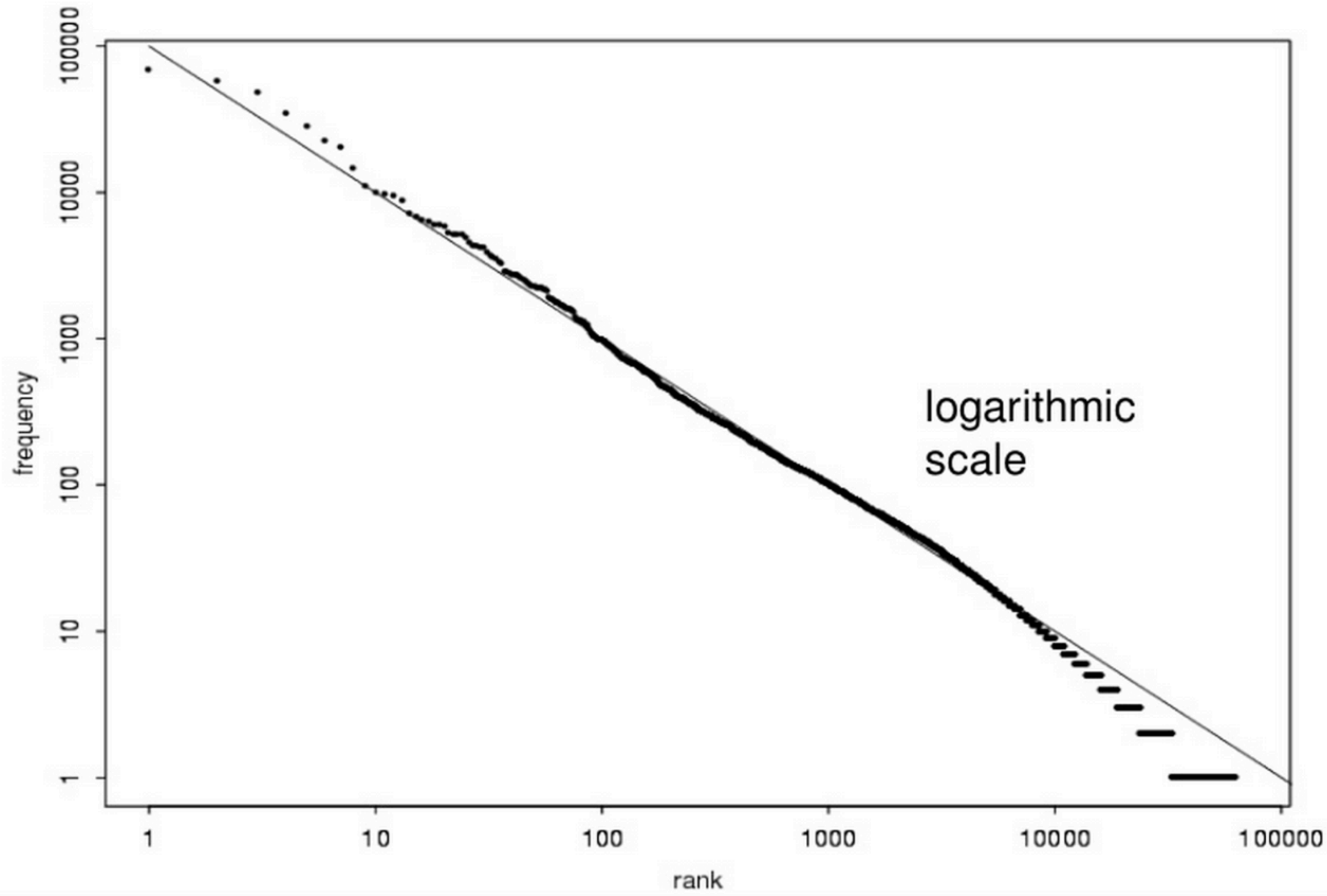
- How it is done:
  - Using simple rules:
    - Remove all punctuation marks (dots, dashes, commas,…)
    - Transform all words to lower case
  - Using a dictionary, such as WordNet, to replace synonyms with a common, often more general, concept
    - E.g., "automobile, car" -> vehicle

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)

ILLUSTRATION OF THE ZIPF'S RULE



Word frequency in the [Brown Corpus](#) of American English text

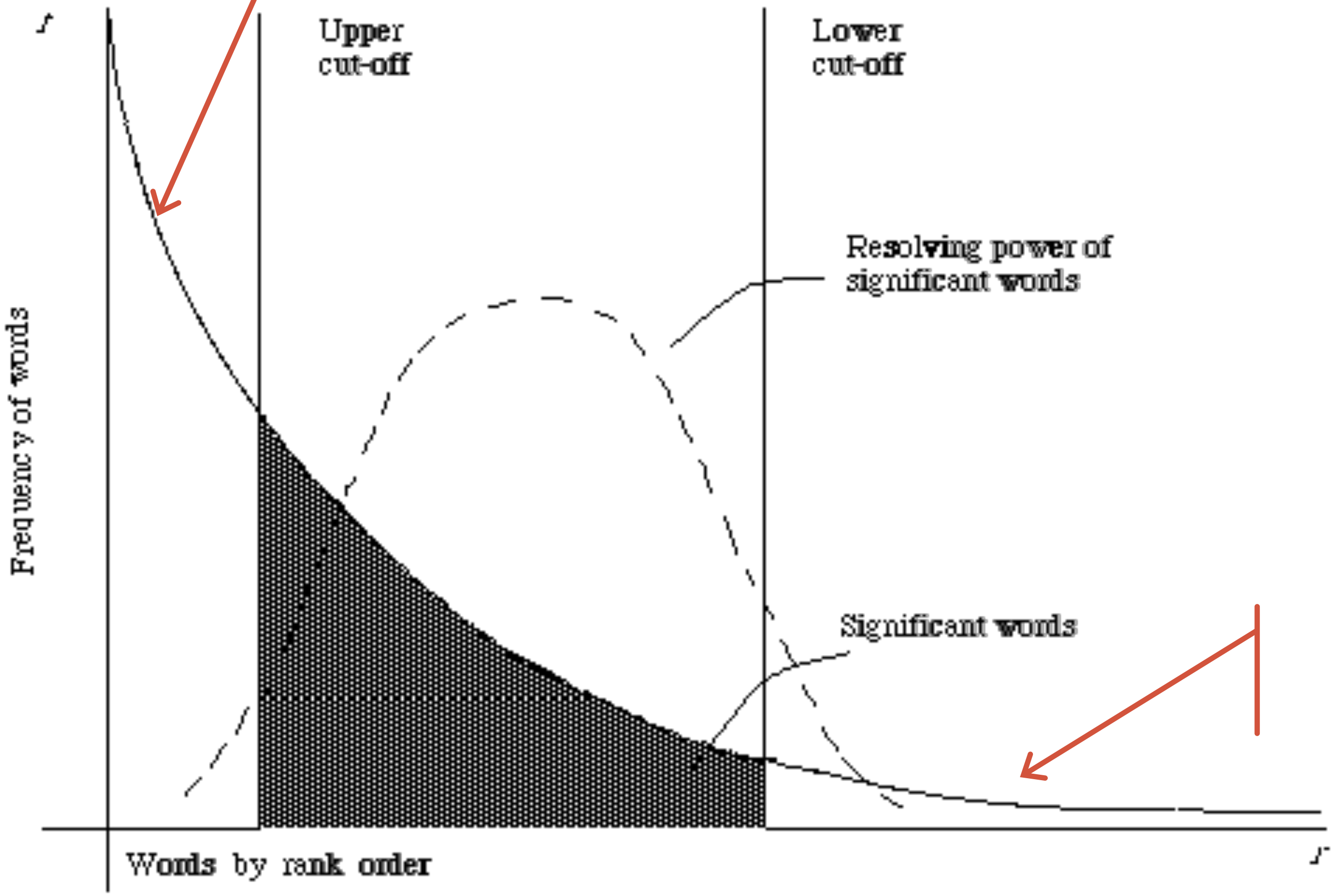
source: http://nlp.stanford.edu/fsnlp/intro/fsnlp-slides-ch1.pdf

IMPLICATIONS OF THE ZIPF'S RULE

- Words in the upper part of the frequency table comprise a significant proportion of all the words in the corpus, but are semantically almost useless
  - Examples: the, a, an, we, do, to
- On the other hand, words towards the bottom of the frequency table are 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

IMPLICATIONS OF THE ZIPF'S RULE

Remove words that do not bear meaning



Remove highly infrequent words

Image source:
http://www.dcs.gla.ac.uk/Keith/Chapter.2/Ch.2.html

STOP-WORDS

- An alternative or a complementary way to eliminate words that are (most probably) irrelevant for corpus analysis
- 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

- Two approaches to decreasing variability of words by reducing different forms of words to their basic / root form
- Stemming is a crude heuristic process that chops off the ends of words without considering linguistic features of the words
  - E.g., argue, argued, argues, arguing -> argu
- Lemmatization refers to the use of a vocabulary and morphological analysis of words, aiming to return the base or dictionary form of a word, which is known as the lemma
  - E.g., argue, argued, argues, arguing -> argue

VSM: COMPUTING TERMS' WEIGHTS

- There are various approaches for determining the terms' weights
- Simple and frequently used approaches include:
  - Binary weights
  - Term Frequency (TF)
  - Inverse Document Frequency (IDF)
  - TF-IDF

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

Example:

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

	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

VSM: Term Frequency

- 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

VSM: Inverse Document Frequency

- The underlying idea: 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:

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

N – number of documents in the corpus

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

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
- General formula for computing TF-IDF:

  TF-IDF(t) = TF(t) x IDF(t)

- One popular 'instantiation' of this formula:

  TF-IDF(t) = tf(t) * log(N/df(t))

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*

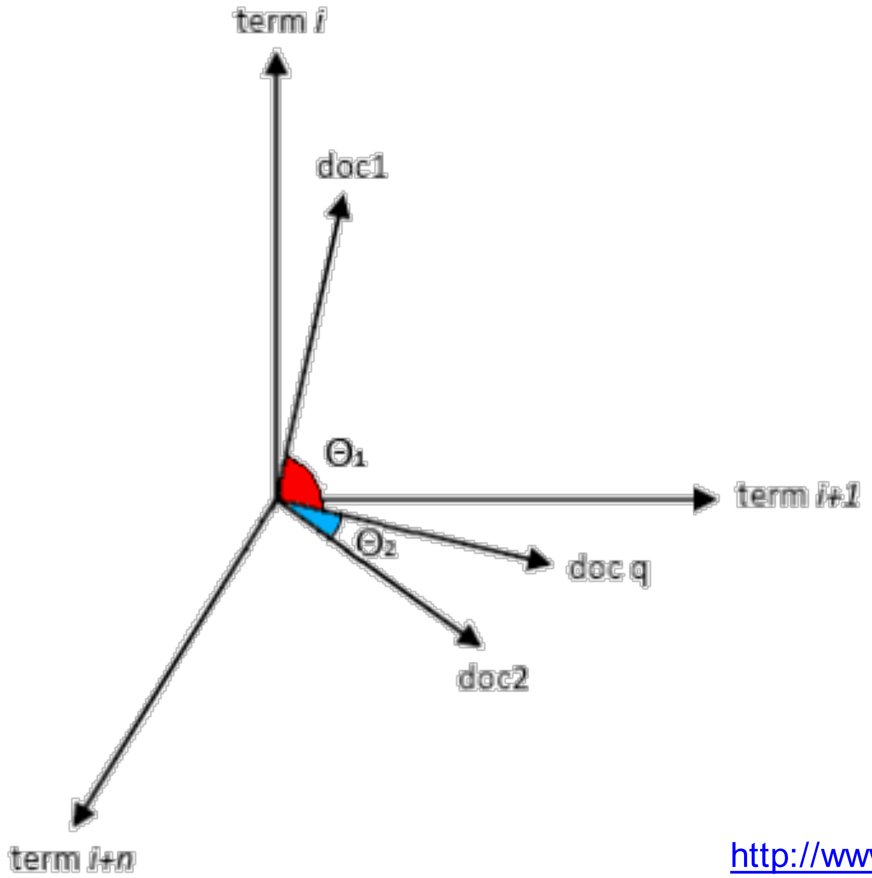


Image source:
http://www.ascilite.org.au/ajet/ajet26/ghauth.html

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

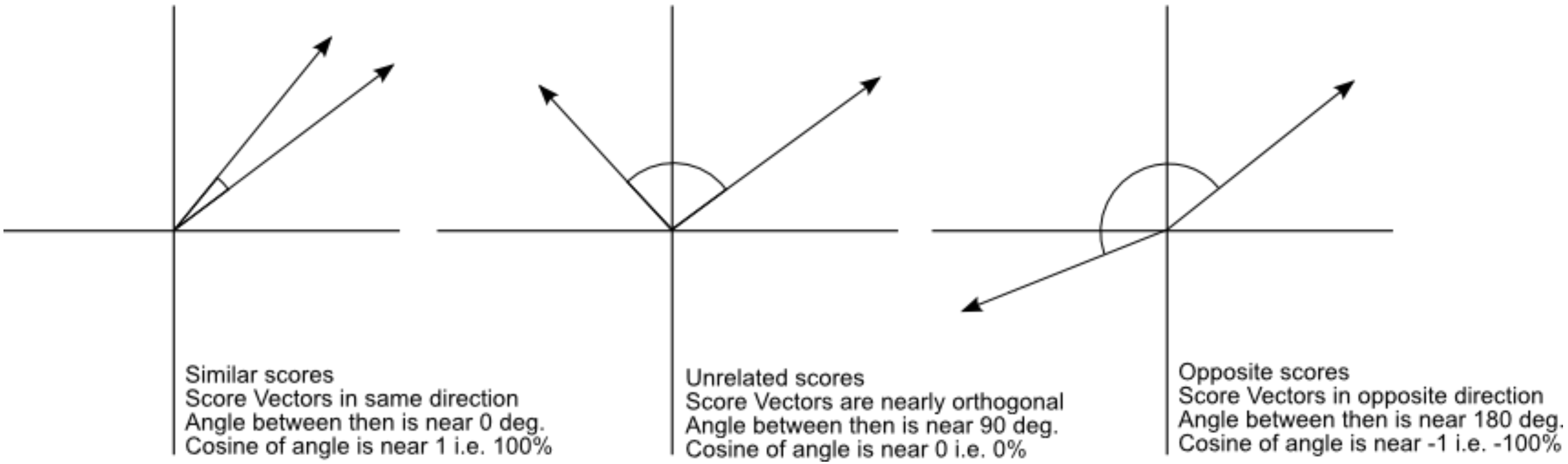


Image source:
http://blog.christianperone.com/?p=2497

VSM: Pros and Cons

- Advantages
  - Intuitive
  - Easy to implement
  - Empirically proven as highly effective
- Drawbacks
  - Based on the unrealistic assumption of words mutual independence
  - Tuning the model's parameters is often challenging and time consuming; this includes selection of method for:
    - determining the terms' weights
    - computing document (vector) similarity

TEXT PROCESSING IN JAVA

Well known and widely used Java frameworks for text processing and analysis:

- Stanford CoreNLP: http://nlp.stanford.edu/software/corenlp.shtml
- Apache OpenNLP: http://opennlp.apache.org/
- LingPIPE: http://alias-i.com/lingpipe/
- GATE: http://gate.ac.uk/

ACKNOWLEDGEMENTS

These slides are partially based on:

- Lecture on Vector Space Model of the Text Mining course @ Uni. of Virginia (link)
- Presentation “Introduction to Text Mining” downloaded from SlideShare.net (link)