Information Retrieval and Vector Space Model

Presenter: Abeer

CSE 6339 - Introduction to Computational Linguistics – Winter 2015
Prof. Nick Cercone

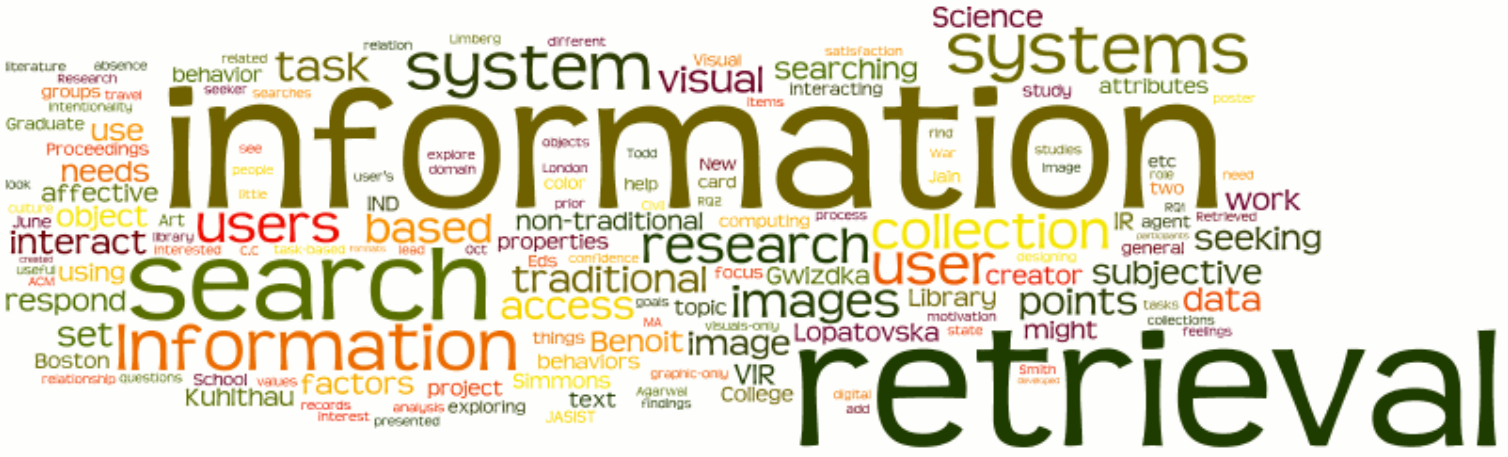
Overview

- What is Information Retrieval (IR)?
- Common Information Retrieval (IR) Tasks
- The Classic Search Model
- Indexing and Inverted Indexing
- Document Representation
- Information Retrieval (IR) Models
- Scoring and Evaluation of Information Retrieval (IR)

What is Information Retrieval (IR)?

- Definition

  "Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."



Why Information Retrieval (IR)?

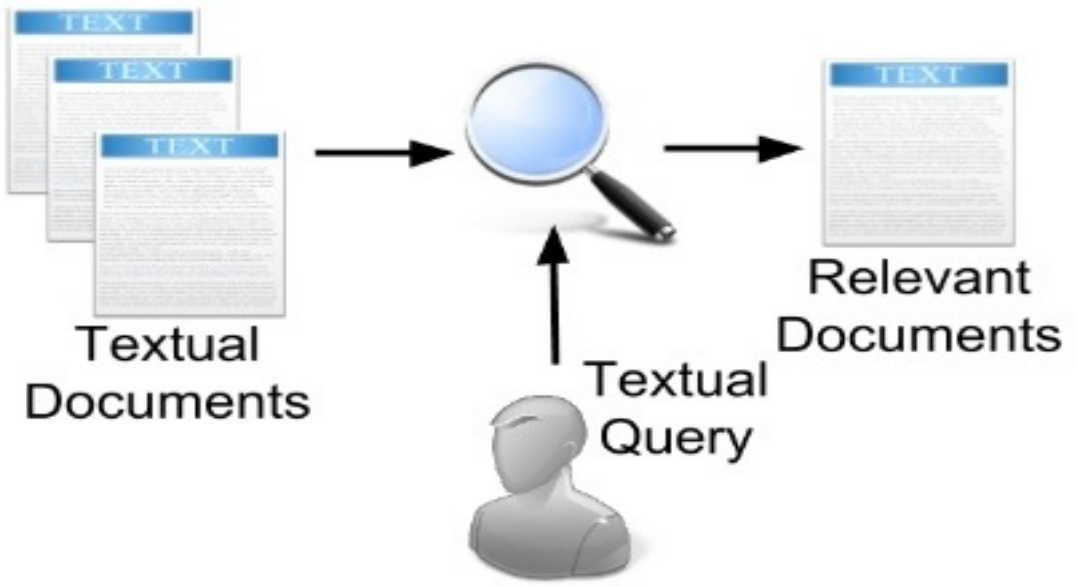
Where we are using Information Retrieval (IR)?

- Automated information retrieval systems are used to reduce what has been called "**information overload**".

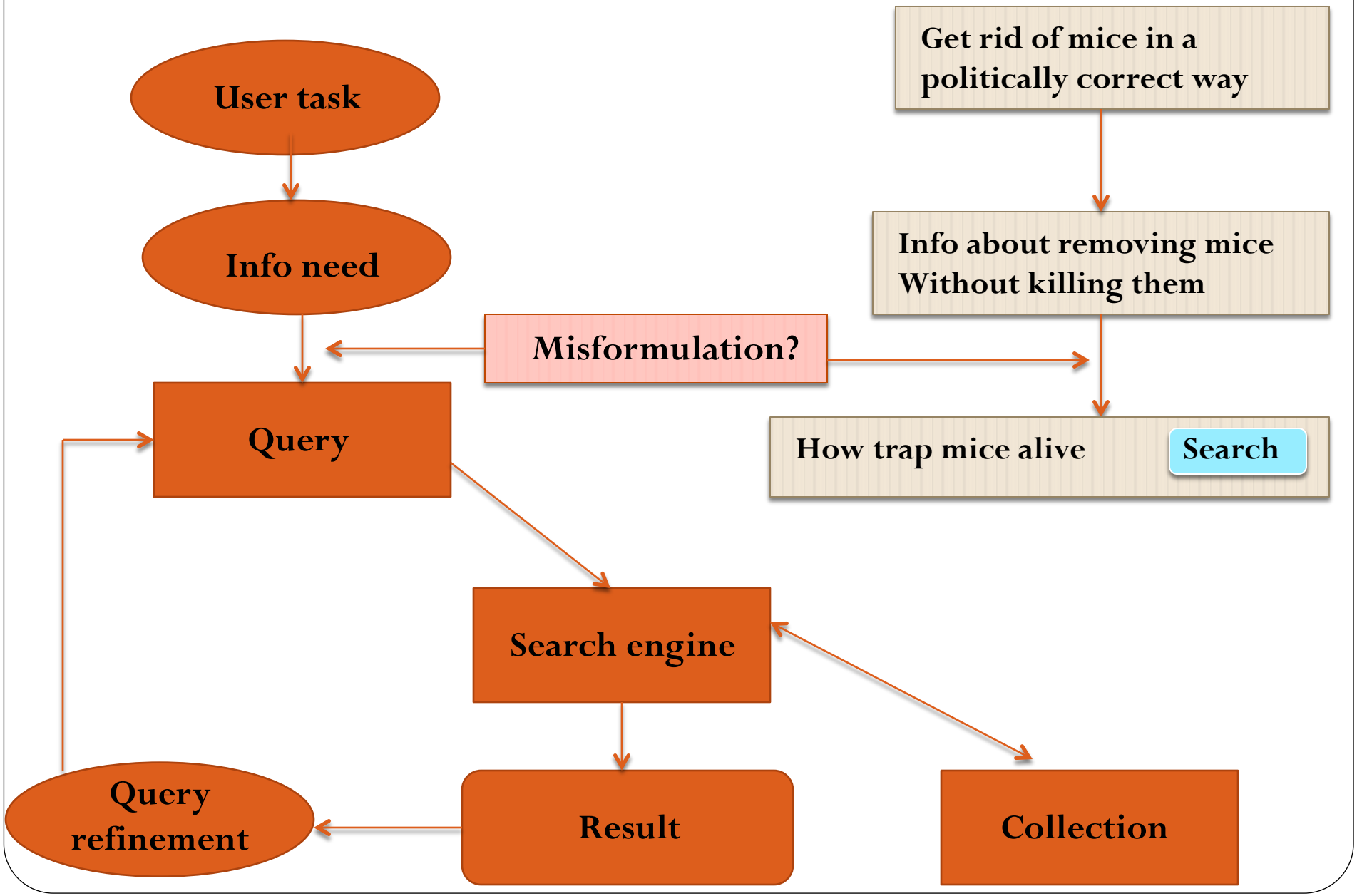
- **Web search engines** are the most visible IR applications.
- These days we frequently think first of web search, but there are many other cases:
  - E-mail search
  - Searching your laptop
  - Many universities and public libraries use IR systems to **provide access** to books, journals and other documents.

Common Information Retrieval(IR) tasks?

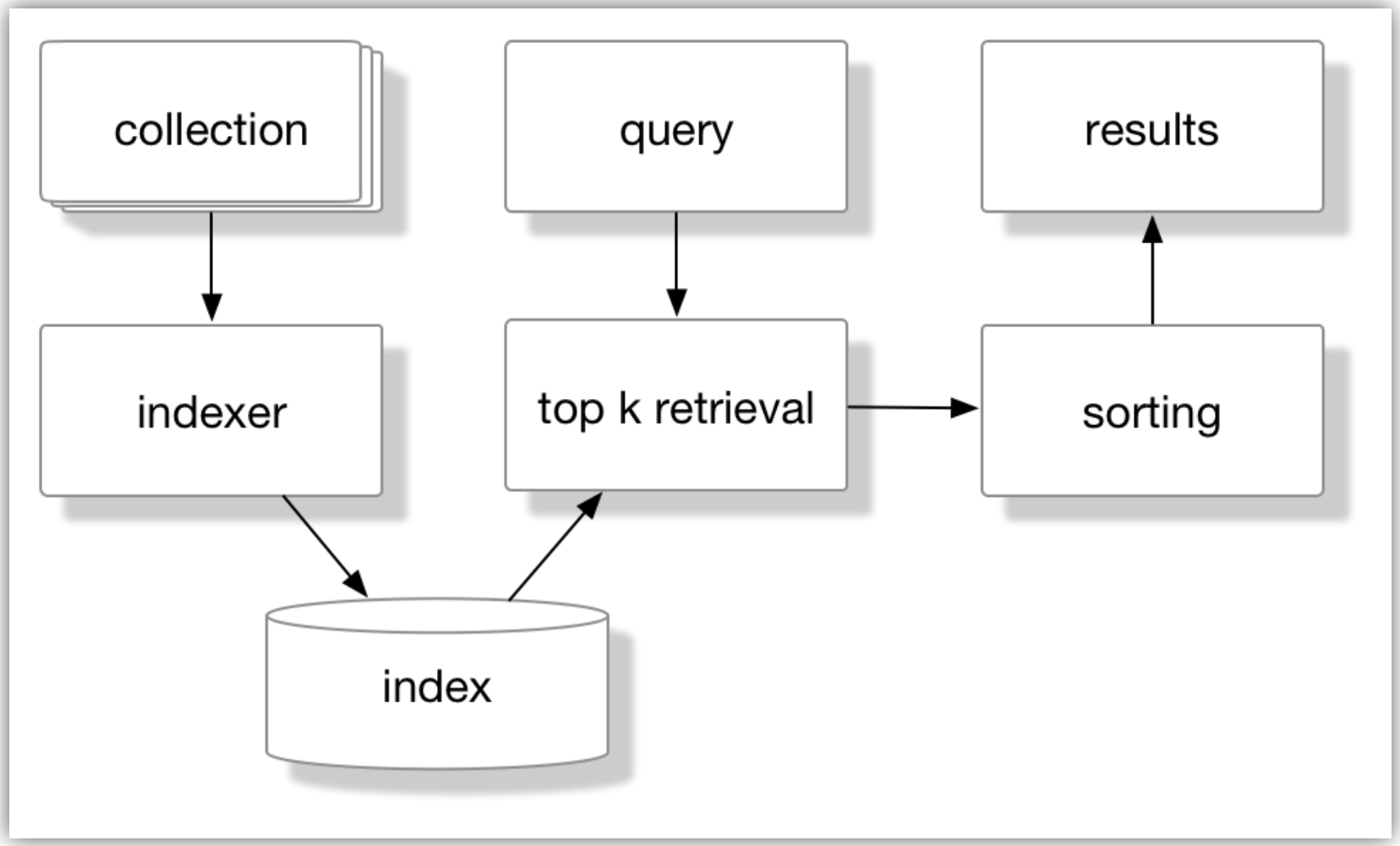
- Common Information Retrieval tasks involve searching for documents relevant to a given query.
- A query contains part of the important words that satisfies info need.



The Classic Search Model



Indexing and Inverted Indexing



Indexing and Inverted Indexing(Cont.)

- Indexing collects and stores data to facilitate fast and accurate information retrieval. Most popular example: Search Engines | Web Search

- Without an index, the search engine would scan every document, which would require considerable time and computing power.

- For example, while an index of 10,000 documents can be queried within milliseconds, a sequential scan of every word in 10,000 large documents could take hours.
  - Need space to store index but it helps in time efficient

Indexing and Inverted Indexing(Cont.)

- This **index** can only determine whether a word exists within a particular document, since it stores no information regarding the frequency and position of the word; it is therefore considered to be a boolean index. Such an index determines which documents match a query but does not rank matched documents.

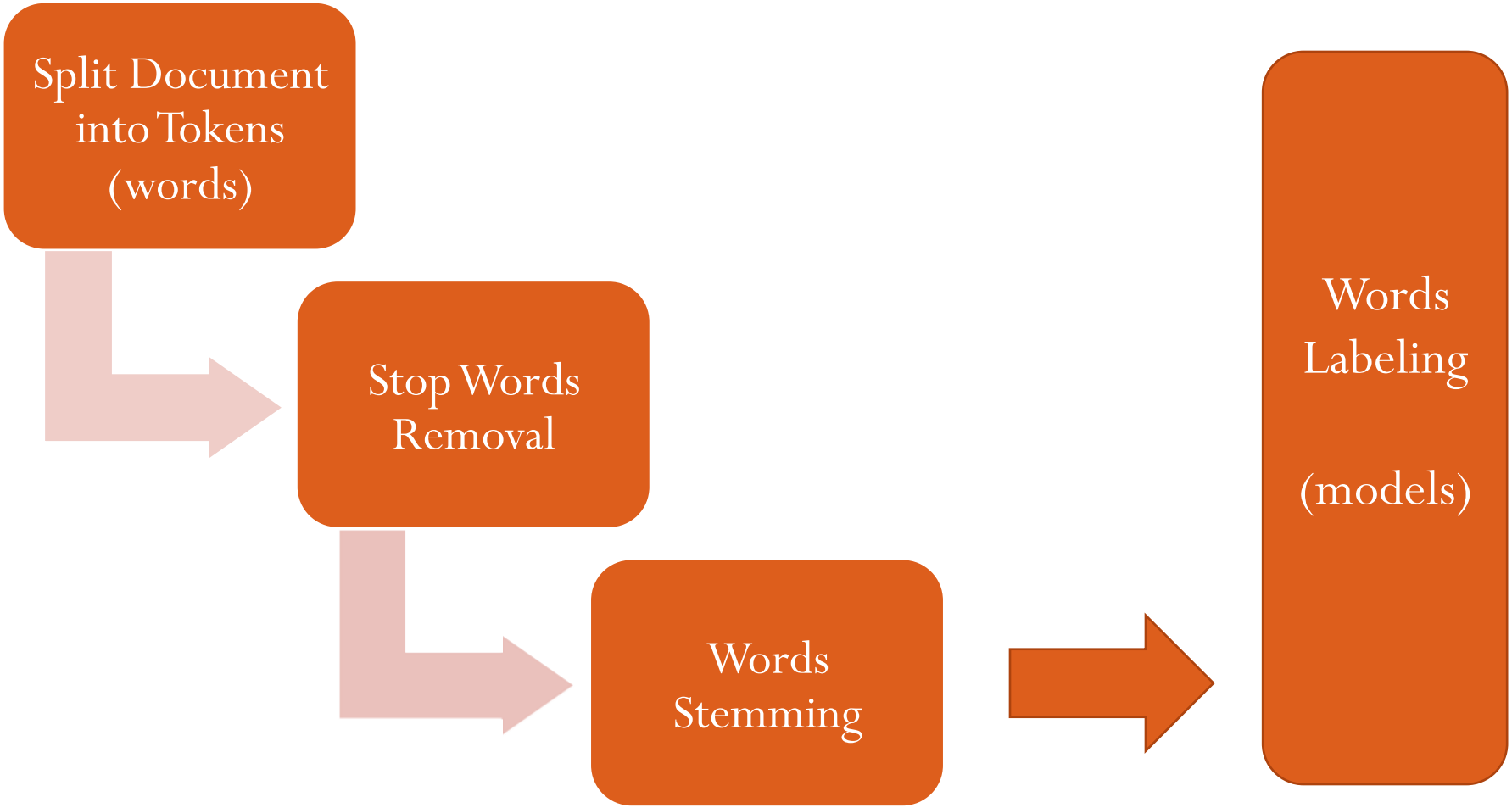
- The **inverted index** is a sparse matrix, since not all words are present in each document.

Indexing and Inverted Indexing(Cont.)

- Many search engines incorporate an **inverted index** when evaluating a search query to quickly locate documents containing the words in a query and then rank these documents by relevance. Because the inverted index stores a list of the documents containing each word, the search engine can use direct access to find the documents associated with each word in the query in order to retrieve the matching documents quickly.

Word	Documents
the	Document 1, Document 3, Document 4, Document 5, Document 7
cow	Document 2, Document 3, Document 4
says	Document 5
moo	Document 7

How to Represent a Document?



Split Document into Tokens (words)

- Extract all the words in a document.

Stop Words Removal

- Many of the most frequently used words in English are worthless in text mining – these words are called *stop words*. Examples of stop words: the, of, and, to, a, …
- Typically about 400 to 500 such words
- These stop words are usually removed from the set of words for representing a document.

Words Stemming

- A technique used to find the root/stem of a word. For example:
  - discussed
  - discusses
  - Discussing
  - Discussion

  Has Stem: **discuss**

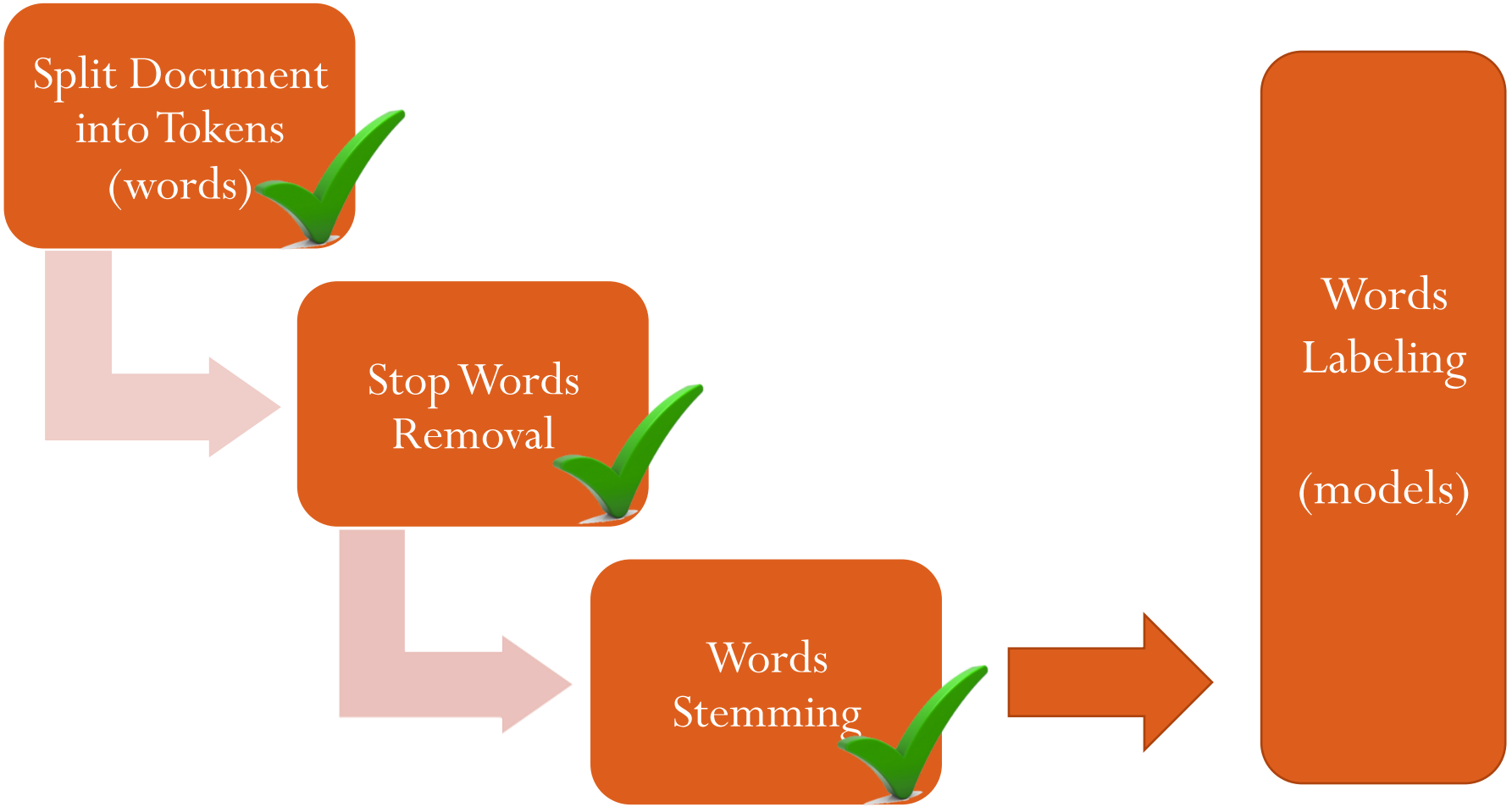
- Usefulness
  - Reduce the number of words
  - Improve effectiveness of text classification

Stemming Algorithm(s)

- **Porter stemming algorithm**

The most widely used stemming algorithm Developed by Martin Porter at the University of Cambridge in 1980

How to Represent a Document?



Information Retrieval (IR) Models

- The documents are typically transformed into a suitable representation.

Some popular and commonly used models are:

- Standard Boolean Model
- Vector Space Model

Standard Boolean Model

- Use 0 or 1 as attribute (word) value to indicate whether the word appears in the document or not.

	Word 1	Word 2	…	Word n
Document 1	1	0	…	1
Document 2	0	1	…	1
…	…	…	…	…
Document n	1	1	…	0

Standard Boolean Model (Cont.)

Documents / words	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

- **Query:** Brutus and Caesar and not Calpurnia

110100
AND **110111**
AND **101111**
100100

The answer to this is:
Antony and Cleopatra and Hamlet

Standard Boolean Model (Cont.)

- Disadvantage :
  - The retrieval results are usually quite poor because (term frequency) is not considered.

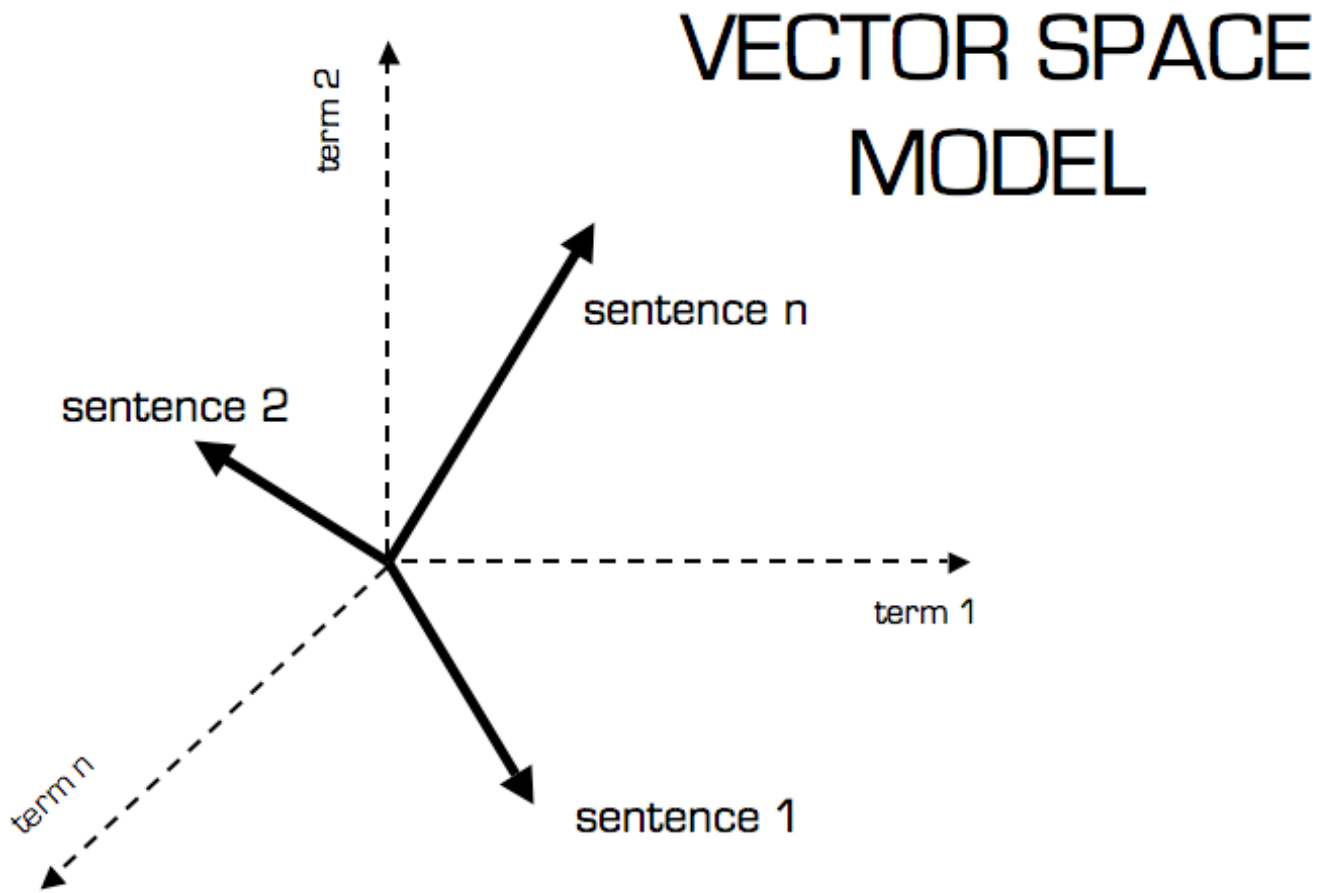
Vector Space Model

- "Vector space model or term vector model is an algebraic model for representing text documents (and any objects, in general) as vectors of identifiers, for example, index terms.
- It is used in information filtering, information retrieval, indexing and relevancy rankings.

$$\vec{d}_i = (w_{i,1}, w_{i,2} \ldots w_{i,t})$$

each $w_{i,j}$ is a weight for term j in document i

Vector Space Model (Cont.)



Vector Space Model (Cont.)

- WEIGHTS

In the classic vector space model proposed by Salton, Wong and Yang, the term-specific weights in the document vectors are products of local and global parameters.

The model is known as **term frequency-inverse document frequency model**.

Term Frequency-Inverse Document Frequency Model

Term Frequency (TF)

- Definition: TF = t_{ij}, frequency of term *i* in document *j*
- Purpose: makes the frequent words *for the document* more important

Term Frequency-Inverse Document Frequency Model (Cont.)

Inverted Document Frequency (IDF)

- Definition: IDF = $\log(N/n\downarrow i)$
  - $n\downarrow i$: number of documents containing term i
  - N : total number of documents
- Purpose: makes rare words *across documents* more important

Term Frequency-Inverse Document Frequency Model (Cont.)

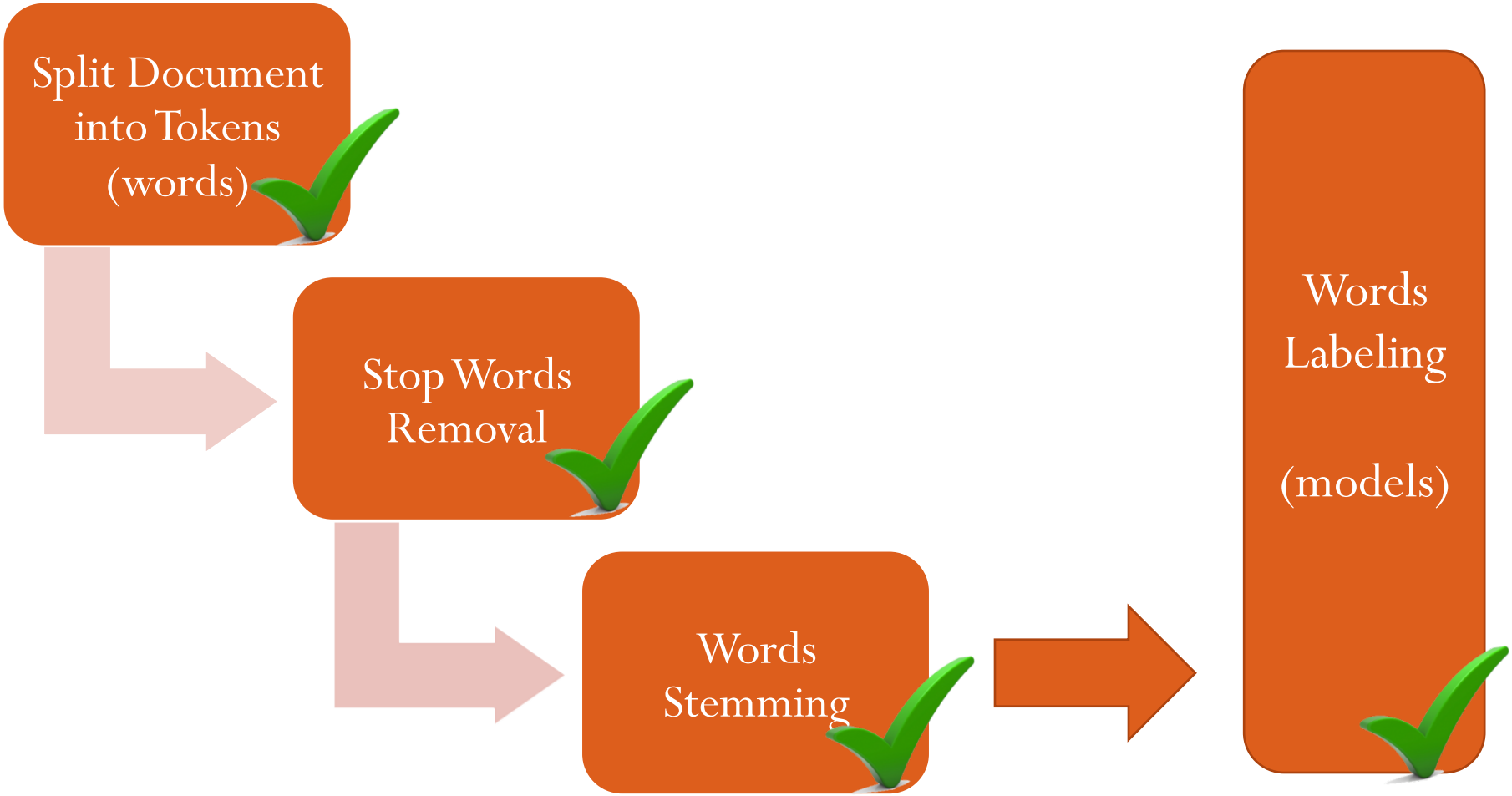
- TF-IDF value of a term i in document j
  - Definition: TF **x** IDF = t_{ij} **x** log(N/n_i)

Term Frequency-Inverse Document Frequency Model - **Example**

- Assume there are three documents in the training set:
  - Document D1: "yes we got no bananas"
  - Document D2: "what you got"
  - Document D3: "yes I like what you got"

	yes	we	got	no	bananas	what	you	I	like
D1:	.18	0.48	0	0.48	0.48	0	0	0	0
D2:	0	0	0	0	0	0.18	0.18	0	0
D3:	0.18	0	0	0	0	0.18	0.18	0.48	.48

How to Represent a Document?



Score as Ranking Function

- Score(q,d):

  The score of a document d is the sum, over all query terms, of the number of times each of the query terms occurs in d.

Score as Ranking Function (Example)

term	query				document			product
	tf	df	idf	$w_{t,q}$	tf	wf	$w_{t,d}$	
auto	0	5000	2.3	0	1	1	0.41	0
best	1	50000	1.3	1.3	0	0	0	0
car	1	10000	2.0	2.0	1	1	0.41	0.82
insurance	1	1000	3.0	3.0	2	2	0.82	2.46

Net score: $0 + 0 + 0.82 + 2.46 = 3.28$

- with $N = 1{,}000{,}000$ documents
- tf-idf as a score: $\text{Score}(q, d) = \sum_{t \in q} tf\text{-}idf_{t,d}$

Evaluation of Information Retrieval

- **How good are the retrieved docs?**

- There are two main mechanisms to evaluate the performance of IR system.
  - *Precision (P)*: fraction of retrieved documents that are relevant.
  - *Recall (R)*: fraction of exact relevant documents.

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})}$$

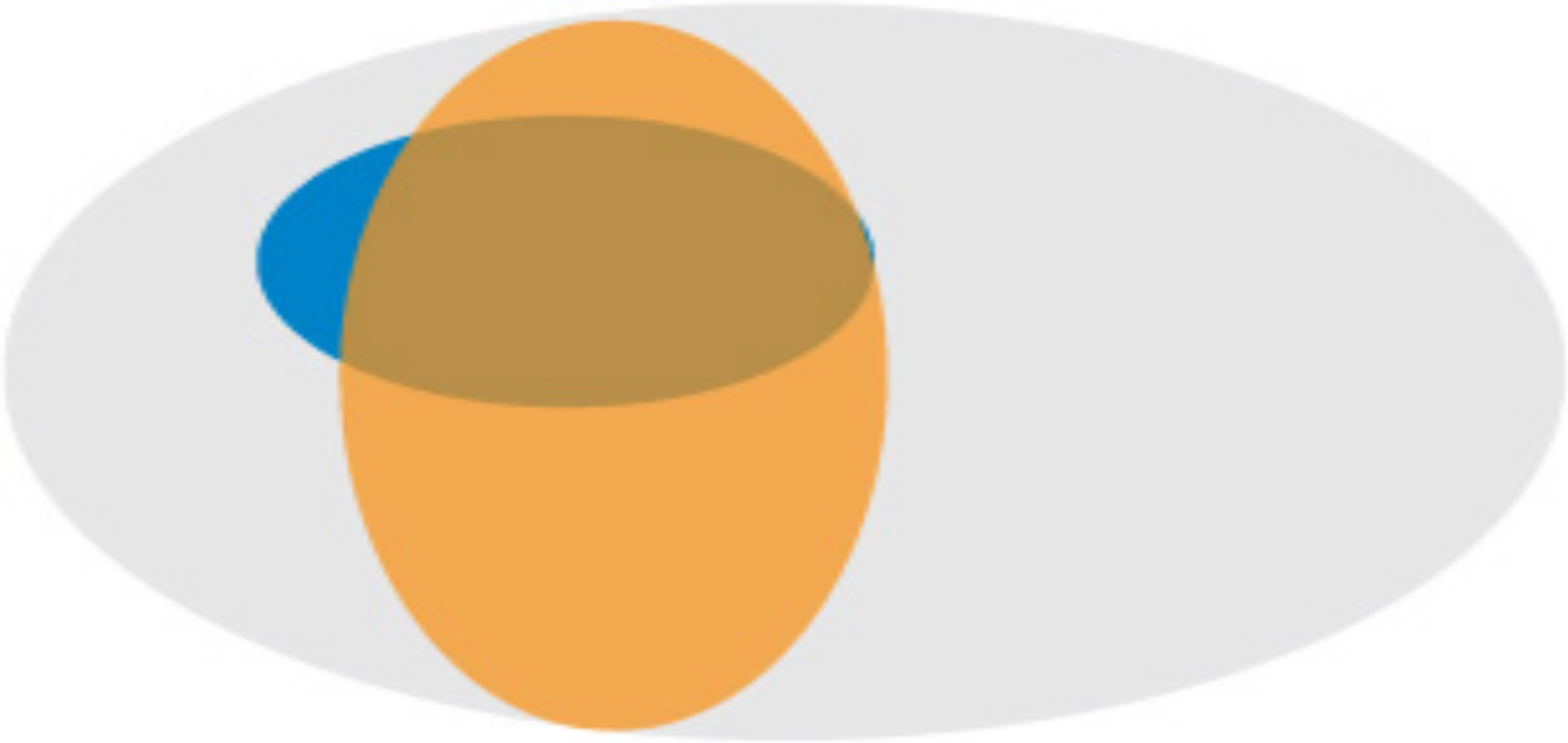
$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})}$$

Evaluation of Information Retrieval (Cont.)

Document Confusion Matrix

	relevant	nonrelevant
retrieved	true positives (tp)	false positives (fp)
not retrieved	false negatives (fn)	true negatives (tn)

- Precision = tp / (tp + fp)
- Recall = tp / (tp + fn)



High recall with low precision is easy to achieve. You can retrieve most of the relevant documents if you cast a net wide enough. But you will also retrieve a lot of what is not relevant.

References

- G. Salton , A. Wong , C. S. Yang, A vector space model for automatic indexing, Communications of the ACM, v.18 n.11, p.613-620, Nov. 1975.
- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. Available at http://www-nlp.stanford.edu/IR-book/.
- Salton G., C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing and Management: an International Journal, v.24 n.5, p.513-523, 1988.
- Salton, G. 1989. Automatic text processing. Chapter 9.

Thank you for your attention!