Knowledge Discovery and Data Mining

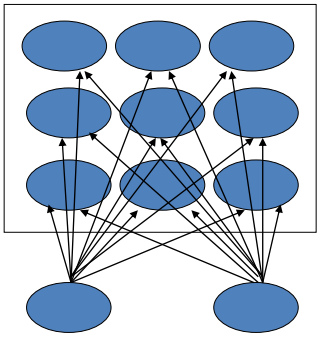
Unit # 9

Kohonen Map

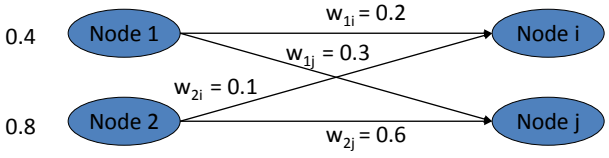
- Kohonen networks are a type of neural network that perform clustering, also known as a knet or a self-organizing map.
- This type of network can be used to cluster the data set into distinct groups when you don't know what those groups are at the beginning.
- Records are grouped so that records within a group or cluster tend to be similar to each other, and records in different groups are dissimilar.
- The basic units are neurons, and they are organized into two layers: the input layer and the output layer (also called the output map).

Kohonen Map (Cont'd)

- Formalized by Teuvo Kohonen in 1982 for unsupervised clustering.
- All of the input neurons are connected to all of the output neurons, and these connections have strengths, or weights, associated with them.
- During training, each unit competes with all of the others to "win" each record.
- Input data is presented to the input layer, and the values are propagated to the output layer. The output neuron with the strongest response is said to be the winner and is the answer for that input.



Working of Kohonen Maps



- The score for classifying a new instance with output node j is given by

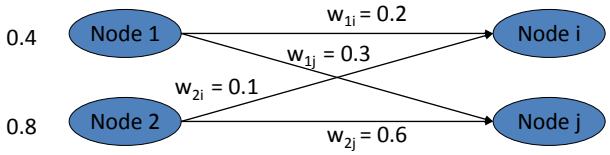
$$\text{sqrt}\left(\sum (n_i - w_{ij})^2\right)$$

  - n_i is the attribute value for the current instance at input i.
  - w_{ij} is the weight associated with the ith input node and output node j.
- Weights are updated according the following formula:

$$w_{ij}\ (\text{new}) = w_{ij}\ (\text{current}) + \Delta w_{ij}$$

  - where $\Delta w_{ij} = r(n_i - w_{ij})$, r is the learning parameter and 0<r<1.

Working of Kohonen Maps (Cont'd)



- Score of Node i: sqrt(0.4-0.2)^2 + (0.8-0.1)^2 = sqrt(0.53)
- Score of Node j: sqrt(0.4-0.3)^2 + (0.8-0.6)^2 = sqrt(0.05)

- Thus, the record belongs to Cluster j.

- Next we update the weights of incoming links to node j. Let r = 0.8
- Δw_{1j} = 0.8 x (0.4 – 0.3) = 0.08
- Δw_{2j} = 0.8 x (0.8 – 0.6) = 0.16
- w_{1j} = 0.3 + 0.08 = 0.38
- w_{2j} = 0.6 + 0.16 = 0.78

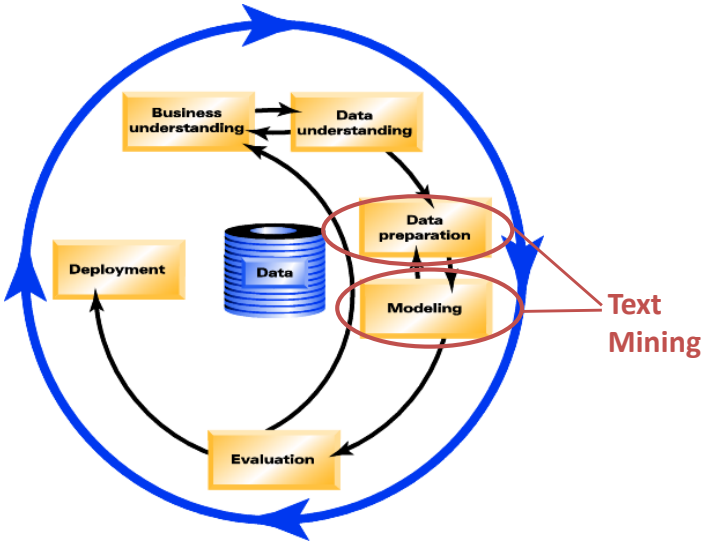
Summary

- Initially, all weights are random. When a unit wins a record, its weights (along with those of other nearby units, collectively referred to as a neighborhood) are adjusted to better match the pattern of predictor values for that record.
- All of the input records are shown, and weights are updated accordingly. This process is repeated many times until the changes become very small.
- As training proceeds, the weights on the grid units are adjusted so that they form a two-dimensional "map" of the clusters (hence the term self-organizing map).

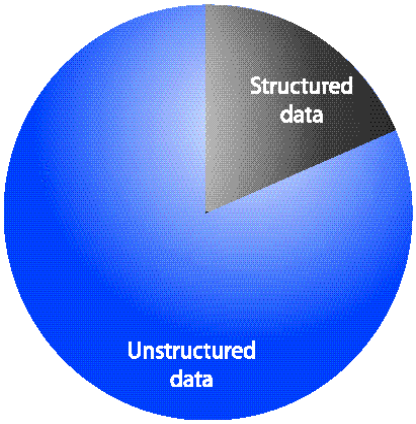
Summary (Cont'd)

- When the network is fully trained, records that are similar should appear close together on the output map, whereas records that are vastly different will appear far apart.
- Usually, a Kohonen net will end up with a few units that summarize many observations (strong units), and several units that don't really correspond to any of the observations (weak units). The strong units (and sometimes other units adjacent to them in the grid) represent probable cluster centers.

Back to the Process



80% of Data is Unstructured



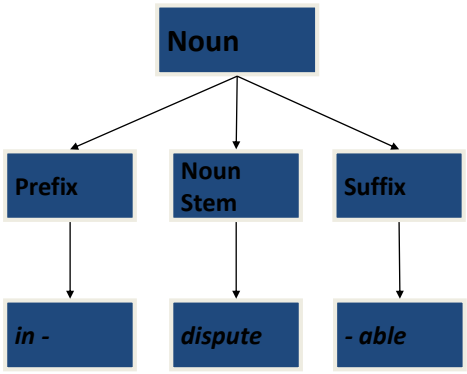
- Database notes:
  - Call center transcripts
  - Other CRM
- Email
- Open-ended survey responses
- Web pages
- NewsGroups
- Social Media

Major Steps in a Text Analytics Process

- Tagging
  - Name Entity Tagging
  - Part-of-Speech Tagging
- Pre-Processing
  - Punctuation Eraser
  - Number Filter
  - N-Char Filter
  - Stop world Filter
- Frequency
- Tag Cloud

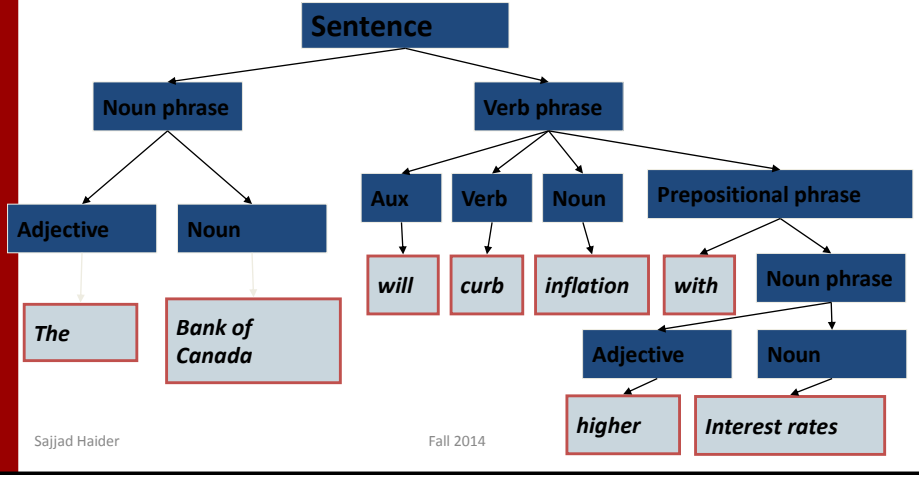
Morphology

- Understanding words
  - Stems
  - Affixes
    - Prefix
    - Suffix
  - Inflectional elements
- Reducing complexity of analysis
- Reduces complexity of representation
- Supports text mining



Syntax

- *The Bank of Canada will curb inflation with higher interest rates*



Part-of-Speech Tagging

a: adjective	b: adverb	c: preposition
d: determiner	n: noun	v: verb
o: coordination	p: participle	s: stop word

How is a Concept Extracted?

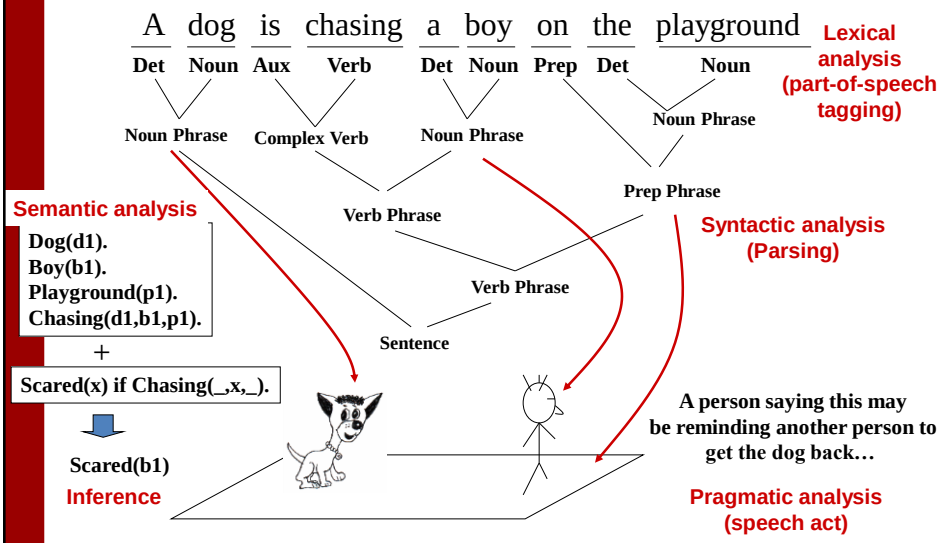
Step 1: Part-of-Speech Tagging

Using	a	tool	like	LexiQuest	Mine	is	a	great
V	P	N	A	N	N	V	P	A

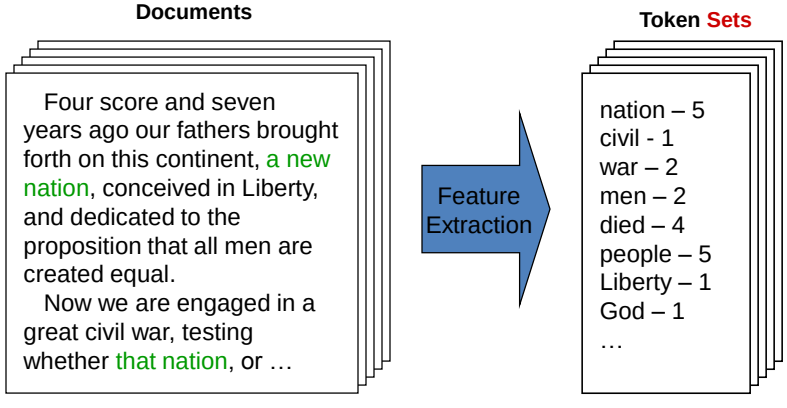
idea	for	any	organization	that	is	interested	in	maintaining
N	P	A	N	P	V	V	P	V

information	on	competitive	intelligence.
N	P	N	N

Natural Language Processing



Bag-of-Tokens Approaches



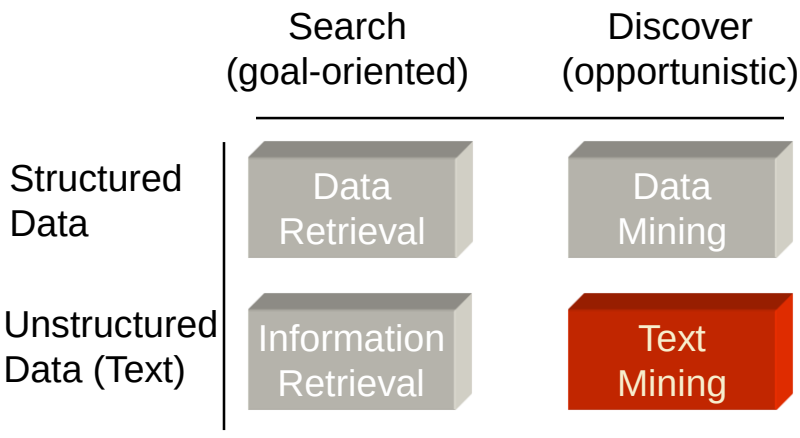
Loses all order-specific information!
Severely limits context!

General NLP—Too Difficult!

- Word-level ambiguity
  - **"design" can be a noun or a verb** (Ambiguous POS)
  - **"root" has multiple meanings** (Ambiguous sense)
- Syntactic ambiguity
  - **"natural language processing"** (Modification)
  - **"A man saw a boy *with a telescope*."** (PP Attachment)
- Anaphora resolution
  - **"John persuaded Bill to buy a TV for *himself*."**
    (*himself* = John or Bill?)
- Presupposition
  - **"He has quit smoking." implies that he smoked before.**

Humans rely on context to interpret (when possible).
This context may extend beyond a given document!

"Search" versus "Discover"



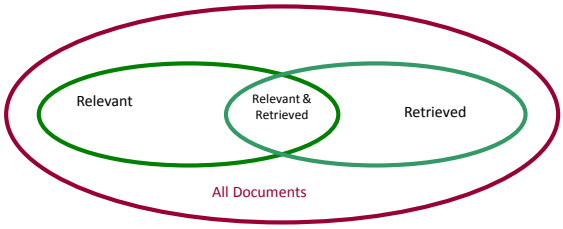
Text Databases and IR

- Text databases (document databases)
  - Large collections of documents from various sources: news articles, research papers, books, digital libraries, e-mail messages, and Web pages, library database, etc.
  - Data stored is usually *semi-structured*
- Information retrieval
  - A field developed in parallel with database systems
  - Information is organized into (a large number of) documents
  - Information retrieval problem: locating relevant documents based on user input, such as keywords or example documents

Information Retrieval

- Typical IR systems
  - Online library catalogs
  - Online document management systems
- Information retrieval vs. database systems
  - Some DB problems are not present in IR, e.g., update, transaction management, complex objects
  - Some IR problems are not addressed well in DBMS, e.g., unstructured documents, approximate search using keywords and relevance

Basic Measures for Text Retrieval



- Precision: the percentage of retrieved documents that are in fact relevant to the query (i.e., "correct" responses)

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|}$$

- Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved

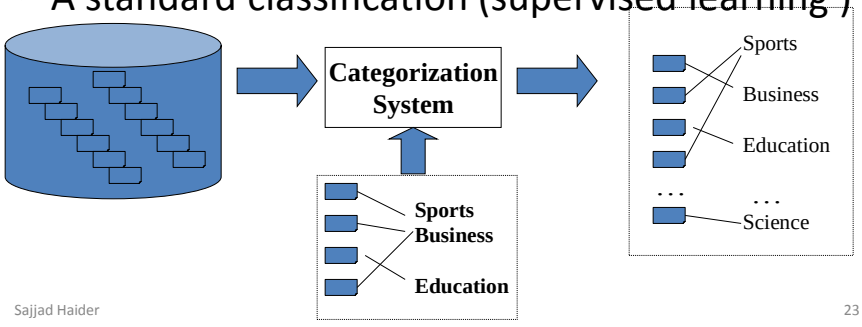
$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}$$

Document Clustering

- Motivation
  - Automatically group related documents based on their contents
  - No predetermined training sets or taxonomies
  - Generate a taxonomy at runtime
- Clustering Process
  - Data preprocessing: remove stop words, stem, feature extraction, lexical analysis, etc.
  - Hierarchical clustering: compute similarities applying clustering algorithms.
  - Model-Based clustering (Neural Network Approach): clusters are represented by "exemplars". (e.g.: SOM)

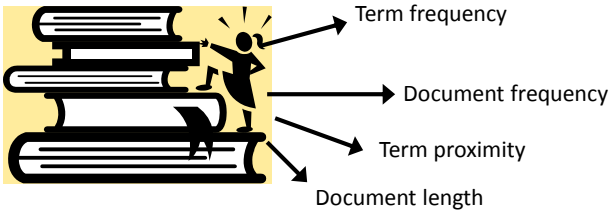
Document Categorization

- Pre-given categories and labeled document examples (Categories may form hierarchy)
- Classify new documents
- A standard classification (supervised learning)



Statistical Analysis

- Use statistics to add a numerical dimension to unstructured text



Computing Relevance

- Call up all the documents that have any of the terms from the query, and count how many times each term occurs:

$$\text{Relevance}_{document} = \sum_{q_i} tf_{q_i}$$

Inverse Document Frequency (idf)

$$idf_i = \log (N/tf_i)$$

- N: Number of documents in corpus
- tf_i: Number of documents in which term occurs in the corpus

- Measures term uniqueness in corpus
  - "phone" vs. "brick"
- Indicates the importance of the term
  - Search (relevance)
  - Classification (discriminatory power)

TF-IDF and Modified Retrieval Algorithm

- Term frequency – inverse document frequency (tf-idf)

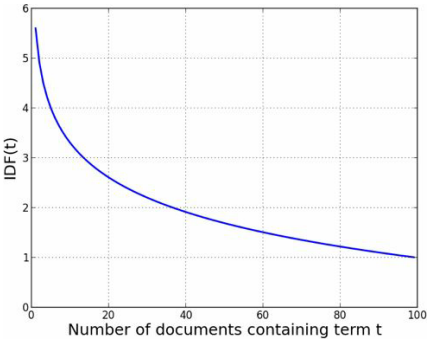
  $tf_{document}$(term) * idf(term)

  query: *"unbrick phone"*
- Document with "unbrick" a few times more relevant than document with "phone" many times
- Measure of Relevance with tf-idf
- Call up all the documents that have any of the terms from the query, and sum up the tf-idf of each term:

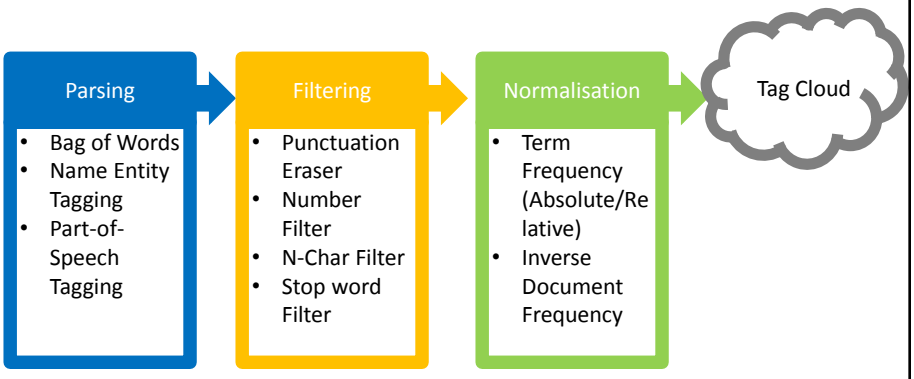
$$\text{Relevance}_{document} = \sum_{q_i} tfidf_{q_i}$$

TF-IDF

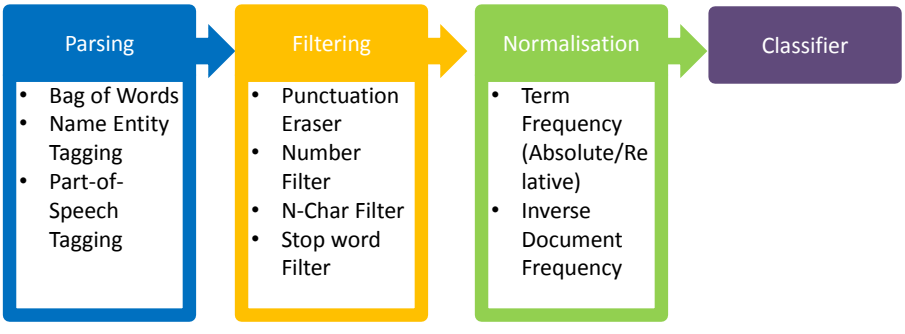
- LESS 'important' words occur MORE often
- Words can be weighted by "their inverse document frequency (IDF)"



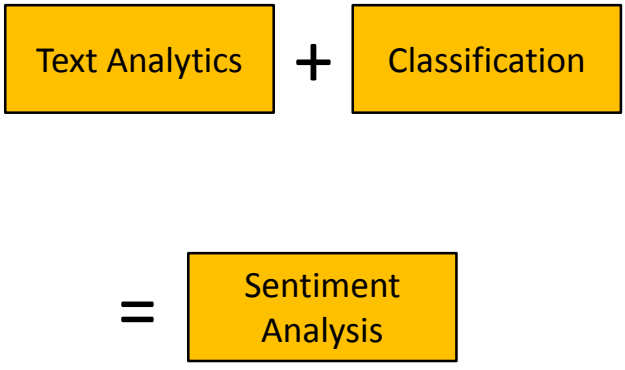
Major Steps



Major Steps



Sentiment Analysis



Tag Clouds

- A **tag cloud** (**word cloud**) is a visual representation for text data, typically used to depict keyword metadata (tags) on websites, or to visualize free form text.
- Tags are usually single words, and the importance of each tag is shown with font size or color.
- This format is useful for quickly perceiving the most prominent terms and for locating a term alphabetically to determine its relative prominence.

TagCrowd.com

action agents algorithm annotation based bayesian best bipedal blend building collaboration comparison coordinated data dynamic evolving extraction files finding fourier frameworks imitation influence information intelligence kick modelling nets networks ontology ontology-based opponent optimization particle portfolio reference-set-based reporting robocup robot rule-based semantic simulation skills soccer sources strategies study testing unstructured via

Wordle.com (1)

Wordle.com (2)

Application of Text Analytics

- Document Classification
  - E-mail Filtering
- Sentiment Analysis
  - Twitter
  - Reviews
- Visualization
- Document Clustering

TEXT ANALYTICS HANDS-ON

Twitter Analysis in KNIME

- Sign up for a Twitter account.
- Log in to Twitter Developers
- Go to My Applications
- Create a new application
- On the application page, click the 'Create my access token' button, wait some seconds and refresh the page
- Enter the following data in the KNIME settings: Consumer key, Consumer secret, Access token, Access token secret