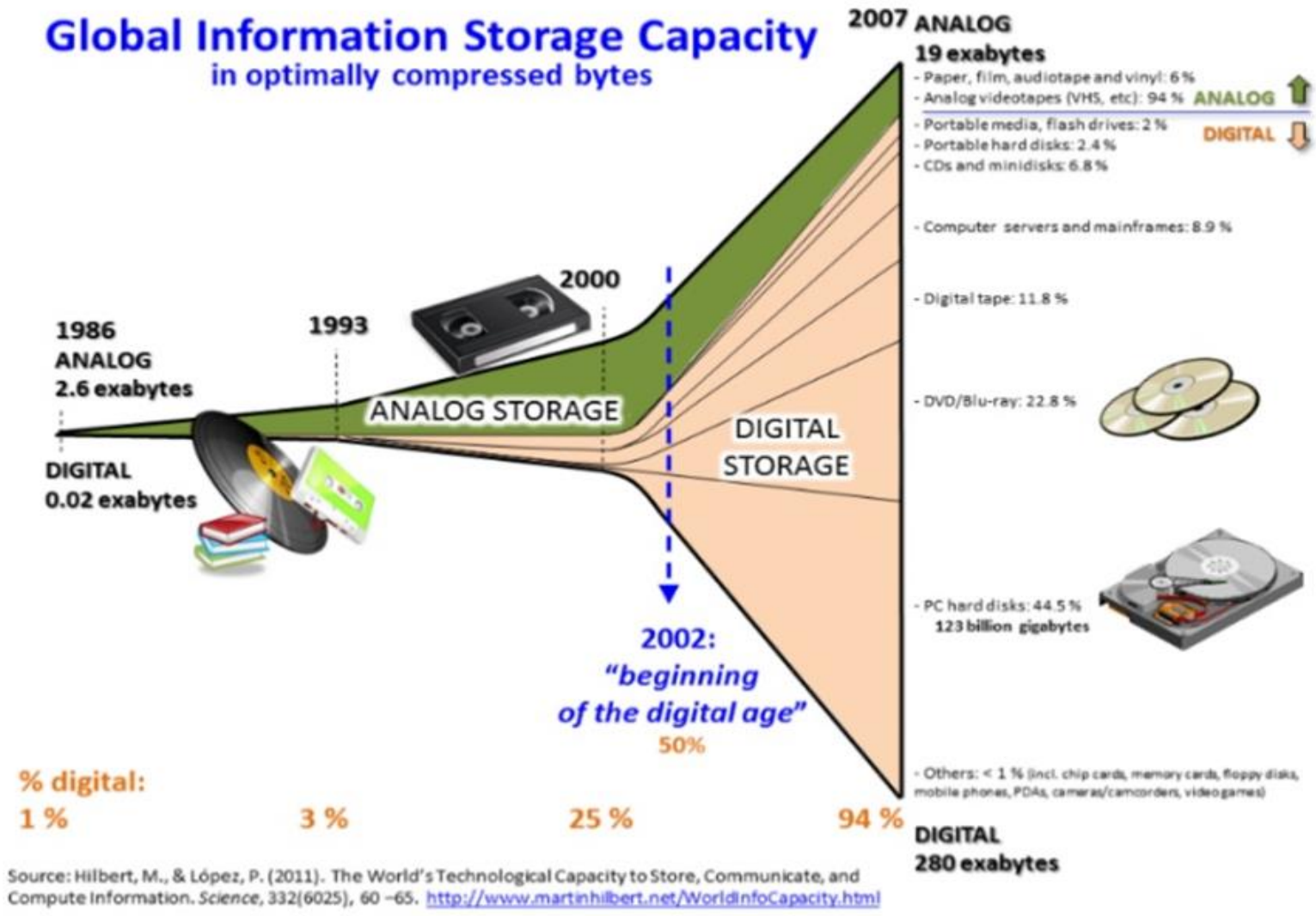
ANALYTIXLABS

Text Mining

Text Mining

Background: Data Explosion in the 21st century

- ✓ Developed economies make increasing the use of data intensive technologies
- ✓ The volume of information can be separated into structured and unstructured data
- ✓ **Structured:** Data is organized in highly mechanized or manageable manner. Some examples include data tables, OLAP cubes, XML formats etc.
- ✓ **Unstructured:** Raw and unorganized data which can be cumbersome and costly to work with. Examples include News Articles, social media, video, Email etc.
- ✓ According to estimates, 80% of world's data is in unstructured text format.



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html

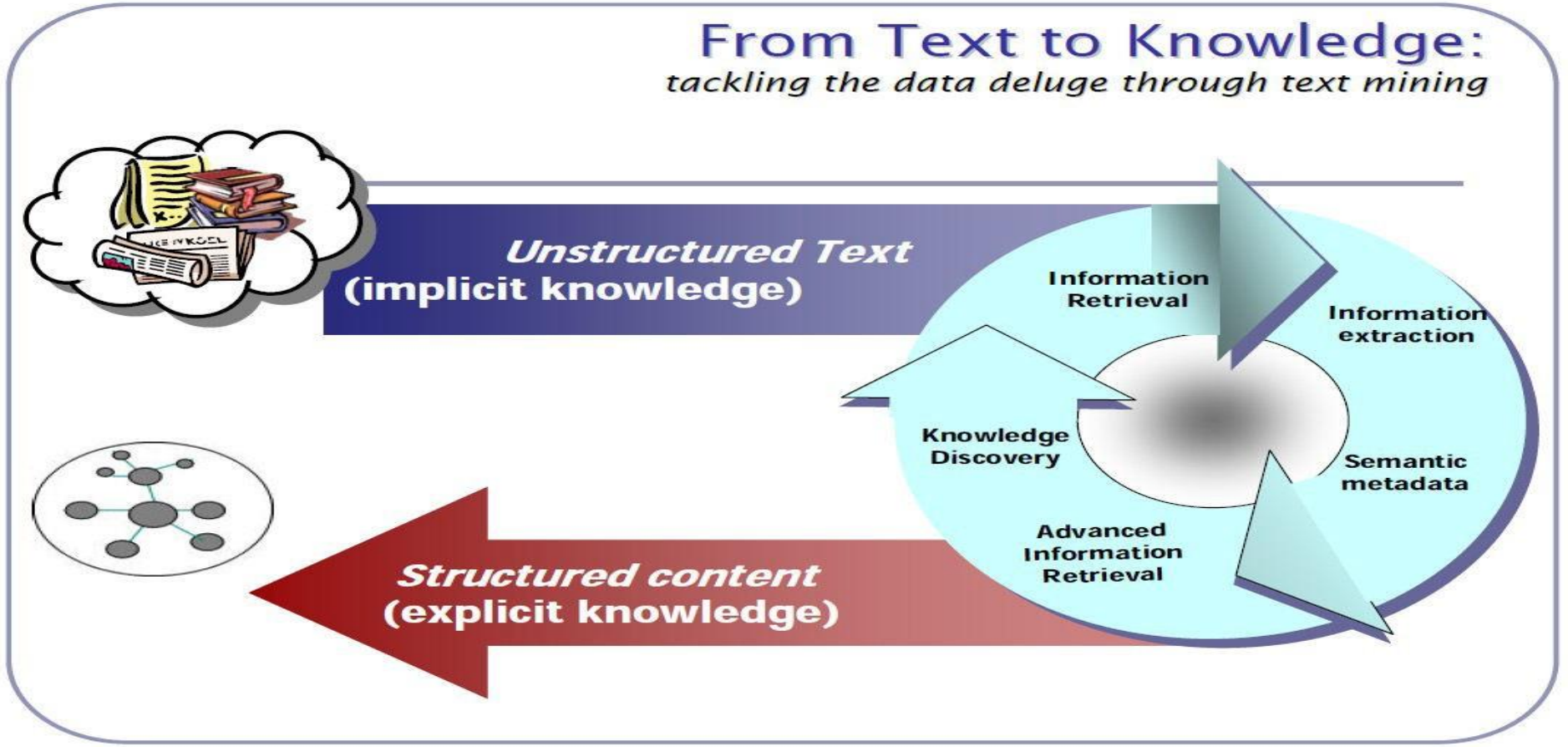
Background: Working with Text data

- ✓ Text data is everywhere – books, news, articles, financial analysis, blogs, social networking, etc.
- ✓ Need methods to extract, summarize, and analyze useful information from this data.
- ✓ Text Mining seeks to automatically discover useful knowledge from the massive amount of data.
- ✓ Lots of research going on in area of text mining in industry and academics.

Background: Working with Text Data

- ✓ The presentation of data for classical data mining and text data mining is quite different.
- ✓ As such, we need to approach text analytics with different techniques in order to bridge the gap between unstructured and structured data realm
- ✓ The main challenge to present the unstructured text correctly in a structured, numerical form
- ✓ We can use some techniques that can be employed when mining with unstructured text data to demystify this process

Text to Knowledge



Features of Text Data & Challenges

- ✓ High dimensionality Large number of features
- ✓ Multiple ways to represent the same concept.
- ✓ Highly redundant data.
- ✓ Unstructured data.
- ✓ Easy for humans, hard for machine. Abstract ideas hard to represent
- ✓ Huge amount of data to be processed.

Features of Text Data & Challenges

Data quality

- **Richness**
  Requires identification of spam, low value content, social media users with multiple accounts etc.
- **Relevance**
  Statistical bias – People posting on social media are not the same as those inactive on the same platform.
- **Copyright legislation**
  Right to copy, digitize and then mine text is severely curtailed by necessary restrictions placed on many text to preserve copyright holders' rights.

Context driven analysis

- **Meaningful dictionary/lexicon for comparison**
  In cases of comparison and classification cases, the result's accuracy depends on the quality of the input classifiers/dictionary.
- **Usefulness of word score**
  Potential ambiguities in the output resulting in high word count of irrelevant words
- **Customization of process flow to suit needs**
  No set process for carrying out text mining. Most software require manual customization of process for conducting different kinds of analysis i.e. word frequency count, classification, clustering, trend analysis etc.

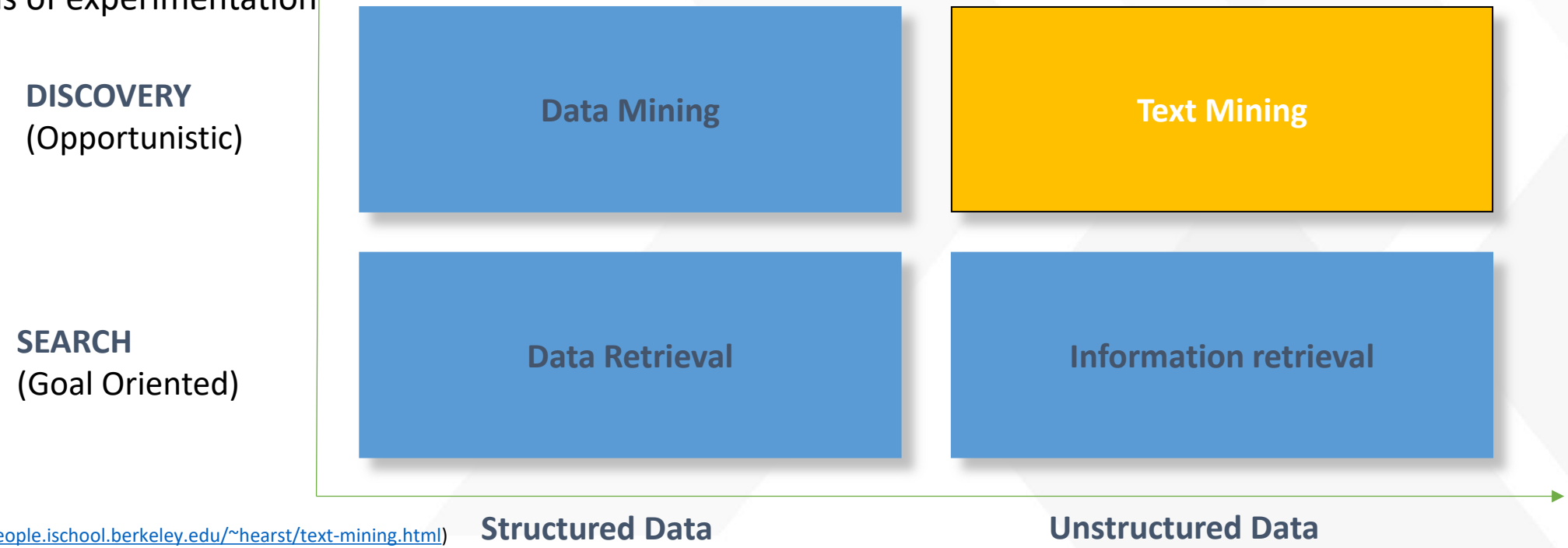
Volume & compatibility

- **Data Volume**
  Data available for mining as a corpus is usually very vast and big in size. This causes issues with storage capacity over time.
- **Software with flexible input formats**
  Text is available in various formats that might not be readable by the current software. Usual readable formats include CSV, XML, excel, txt file etc. Web crawled or other available data may not be compatible.

What is text mining?

Text Mining and Text Analytics are broad umbrella terms describing range of technologies for analyzing and processing semi-structured and unstructured data

Text Mining is the discovery (by a computer) of new, previously unknown information, by automatically extracting and relating information from different unstructured textual resources, to reveal otherwise hidden meanings. A key element is the linking together of the extracted information to form new facts or new hypotheses to be explored further by more conventional means of experimentation[1]



1 Marti Hearst, U C Berkeley (http://people.ischool.berkeley.edu/~hearst/text-mining.html)

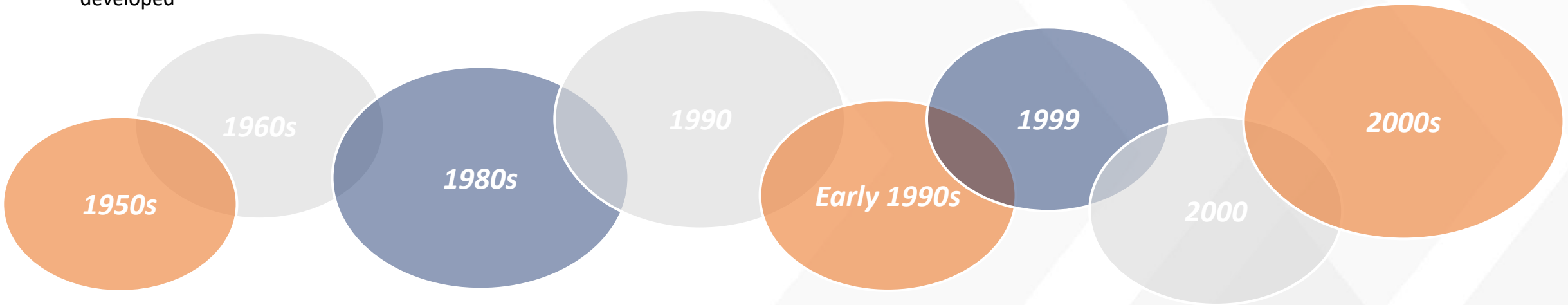
Major milestones in text mining through the years …

- Early findings of the Cranfield studies, developing a model for **Information Retrieval** system evaluation.
- National Library of Medicine developed MEDLARS **Medical Literature Analysis and Retrieval**
- **Management Information Systems** developed

- The **emergence of text analytics as Text Data Mining** stemmed from a refocusing of research going from algorithm development to application
- Revolutionary paper on **Untangling Text Data Mining** by Prof. Marti A. Hearst

- **Implementation of Semantic Indexing** technology for intelligence community for analyzing unstructured text (SAIC).

- **Commercial text mining software** surfaced, having multiple functionality – better visualizations, multi lingual capability
- With the **advent of social media** text mining increasingly applies for semantic search, customer intelligence and news analysis



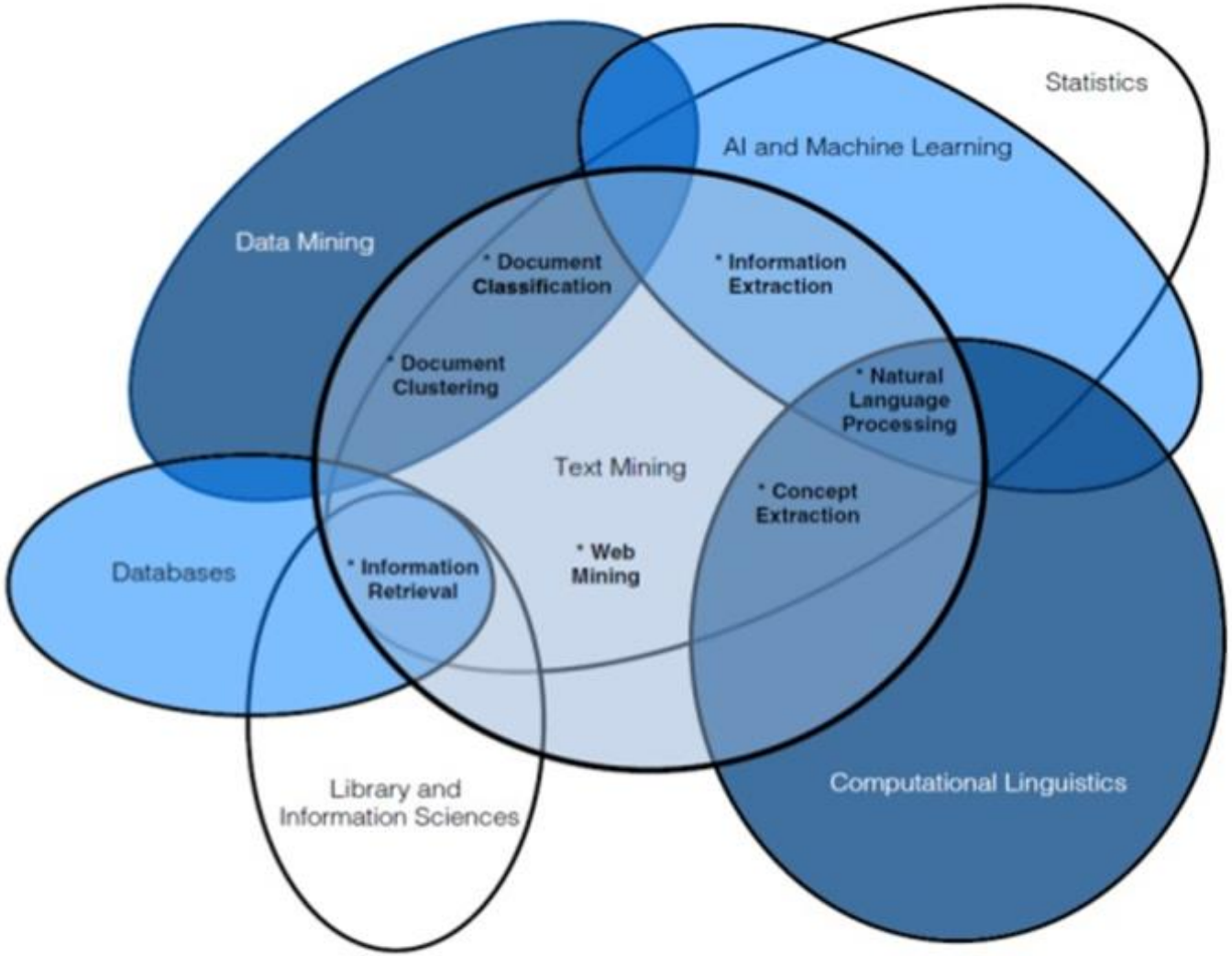
- **Business intelligence first defined as** utilizing data-processing machines for abstracting and encoding of documents and creating interest profiles for each of the action points in the organization

- **Labor-intensive manual text mining** approaches first surfaced
- Emergence of **Business Intelligence**
- A revolution in Natural Language Processing with the **introduction of machine learning algorithms** for language processing.

- **First TREC (Text Retrieval)** conference
- Patent granted for the **cross-lingual application** of Latent Semantic Indexing
- **Web search engines implementation** of many features formerly found only in experimental **Information Retrieval** systems

- Various **standard algorithms for stemming and information retrieval** developed by Dr. Porter
- After two decades of numbers-focused business intelligence, **analytical tools and techniques – reporting, OLAP, data mining, ETL and data warehousing** were widely adopted

Application areas



Application areas

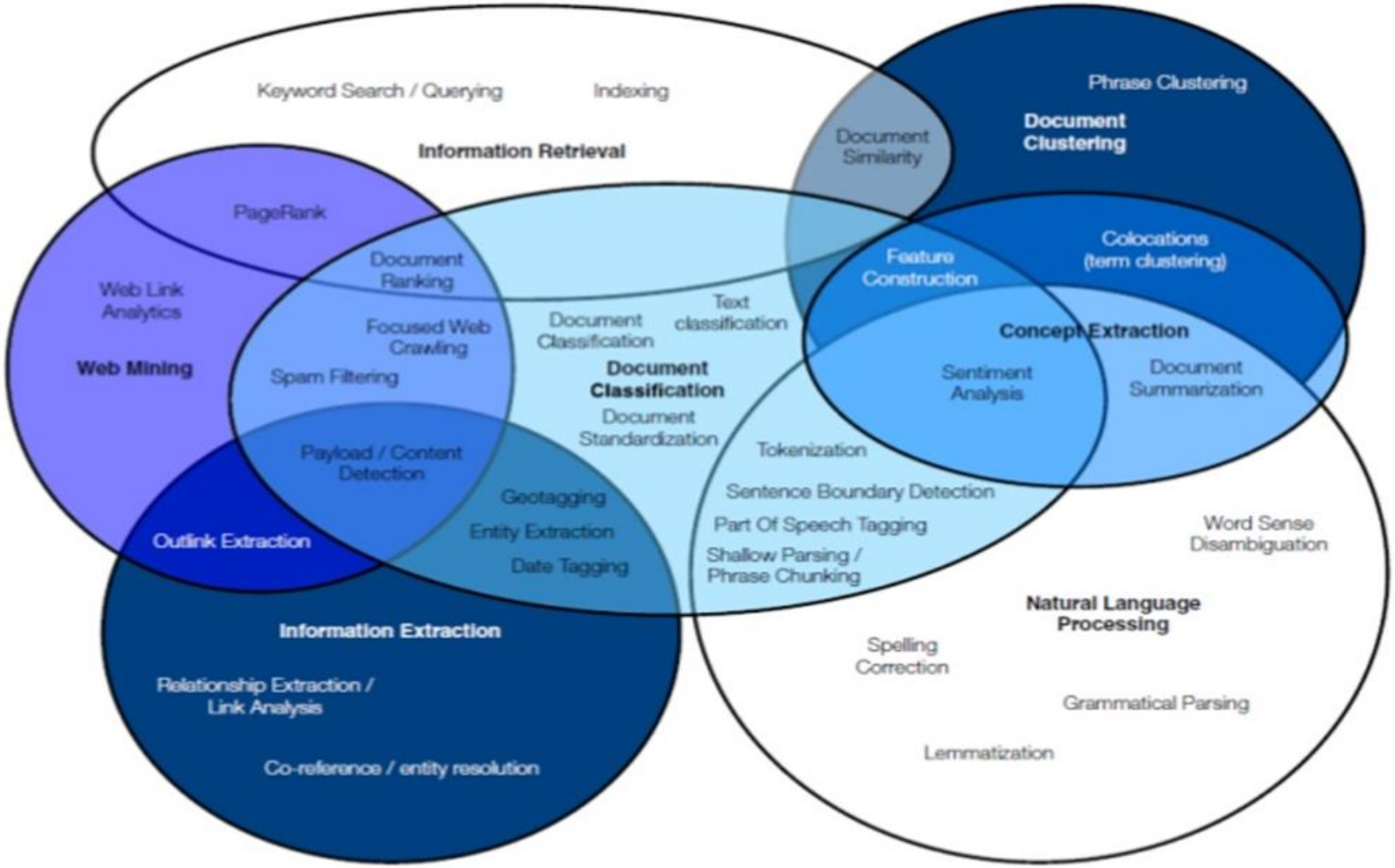
- ✓ Search and Information Retrieval (IR): Storage and retrieval of text documents, including search engines and keyword search
- ✓ Document Clustering: Grouping and categorizing terms, snippets, paragraphs or documents using data mining clustering methods
- ✓ Text Categorization/Document classification: Grouping and categorizing snippets, paragraphs, or documents using data mining classification methods, trained or labelled examples
- ✓ Web Mining: Data and text mining on the internet, with specific focus on the scale and interconnectedness of the web
- ✓ Information Extraction (IE): Identification and extraction of relevant facts and relationships from unstructured and semi structured text
- ✓ Natural Language process (NLP): Low-level language processing and understanding tasks (Ex: Tagging parts of speech); often used synonymously with computations linguistics
- ✓ Concept Extraction: Grouping of words or phrases into semantically similar groups
- ✓ Discovering associations between terms
- ✓ Generating hypothesis
- ✓ Ssummarizing large amount of textual and factual data.
- ✓ Link analysis

Major Areas of Text Analytics

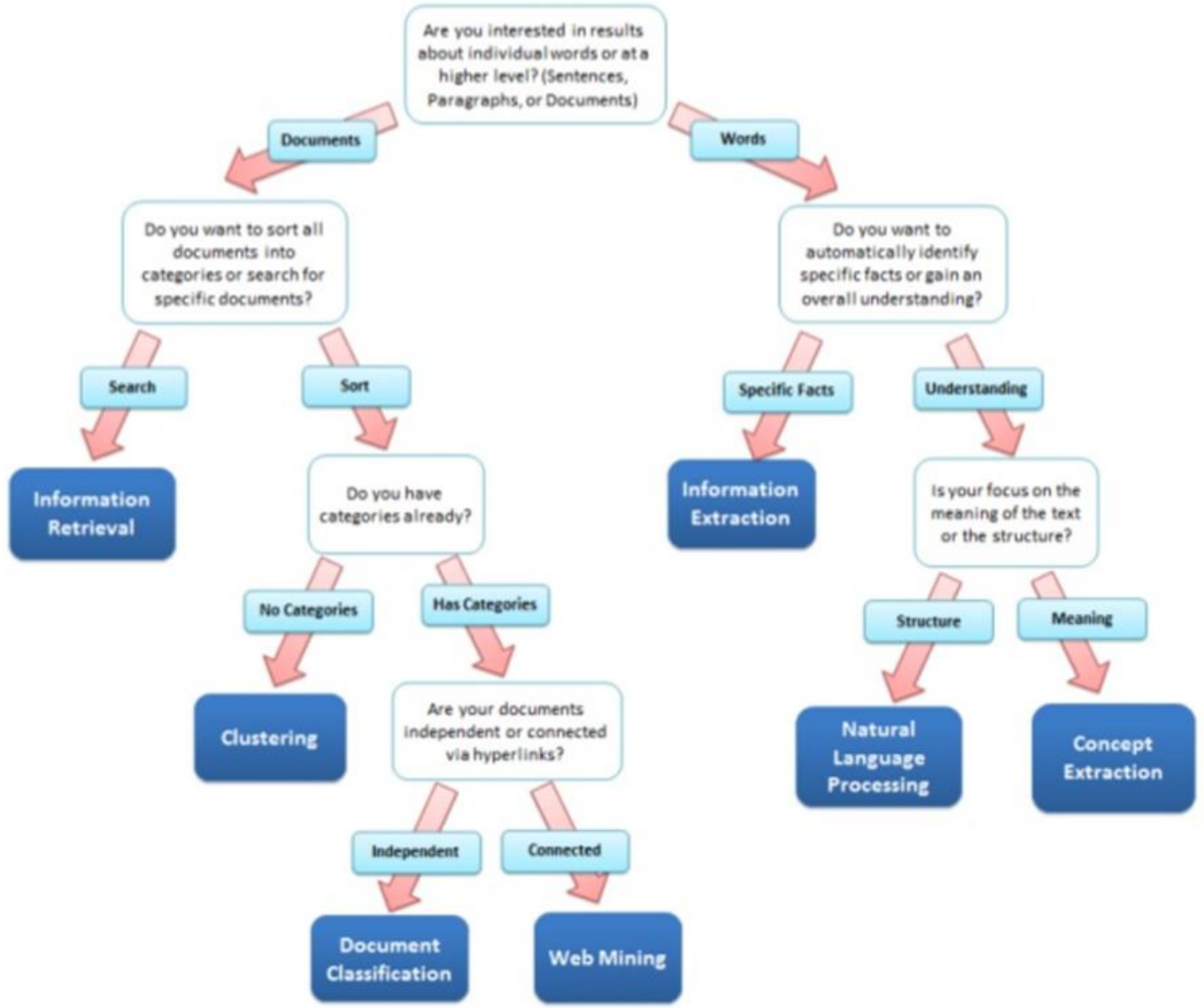
Finding a Practice Area Based on the Desired Product of Text Mining	
Desired Application	**Practice Area**
Linguistic Structure	Natural Language Processing
Topic / Category Assignment	Document Classification
Documents that match keywo	Information Retrieval
A structured database	Information Extraction
"Needles in a Haystack"	Document Classification
List of synonyms	Concept Extraction
Marked Sentences	Natural Language Processing
Understanding of microblogs	Web Mining
Similar documents	Document Clustering

Text Mining Topics and Related Practice Areas	
Topic	**Practice Area**
Keyword Search	Search and Information Retrieval
Inverted Index	Search and Information Retrieval
Document Clustering	Document Clustering
Document Similarity	Document Clustering
Feature Selection	Document Classification
Sentiment Analysis	Document Classification
Dimensionality Reduction	Document Classification
eDiscovery	Document Classification
Web Crawling	Web Mining
Link Analytics	Web Mining
Entity Extraction	Information Extraction
Link Extraction	Information Extraction
Part of Speech Tagging	Natural Language Processing
Tokenization	Natural Language Processing
Question Answering	Natural Language Processing
Topic Modeling	Concept Extraction
Synonym Identification	Concept Extraction

Major Areas of Text Analytics



Identifying the text mining task



Text Mining Basics

- ✓ Text is Unstructured collections of words
- ✓ Documents are basic units consisting of a sequence of tokens or terms
- ✓ Terms are words or roots of words, semantic units or phrases which are the atoms of indexing
- ✓ Repositories (databases) and corpora are collections of documents.
- ✓ Corpus conceptual entity similar to a database for holding and managing text documents
- ✓ Text mining involves computations to gain interesting information

Text Mining Frame work & Process

CONCEPTUAL PROCESS:

- ✓ organize and structure the texts (into repository)
- ✓ convenient representation (preprocessing)
- ✓ Transform texts into structured formats (e.g. TDM)

FRAME WORK:

Different file formats and in different locations

- ✓ standardized interfaces to access the document (sources)

Metadata valuable insights into the document structure

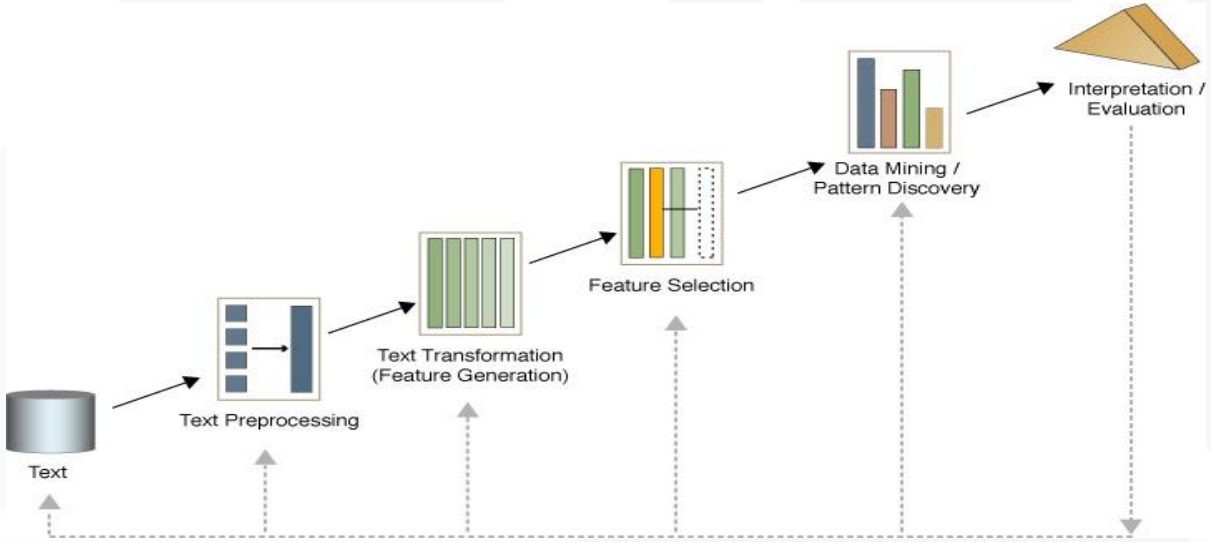
- ✓ must be able to alleviate metadata usage

To efficiently work with the documents

- ✓ must provide tools and algorithm to perform common task (transformation)
- ✓ To extract patterns of interest (filtering)

TEXT MINING PROCESS:

- **Text preprocessing**
  - Syntactic/Semantic text analysis
- **Features Generation**
  - Bag of words, Vector Space
- **Features Selection**
  - Simple counting Statistics
- **Text/Data Mining**
  - Clustering- Unsupervised learning
  - Taxonomy/Classification
  - Predictive Modeling
- **Analyzing results**



How is text mining done?

Collection	Pre-processing	Transformation (Feature generation/selection)	Using algorithms (Text/Data Mining)	Evaluation (Analyzing results)
Unstructured data for our analyses can be found in these various forms : • Text files/CSV • JSON files • HTML files • Excel files • Websites The objective is to be able to capture and store our data without compatibility issues. In most cases, the data is readily available (except for data stored on the Web, which needs to be crawled)	Once the data has been loaded, it is vital to make it ready for suitable analyses. A few generic pre-processing steps are : • Tokenizing text • POS tagging • Disambiguation • Cleanup of non-textual data (white space, numbers, symbols)	Transformation would ensure that the text corpus is enriched and homogenous. These processes include : • Stop word removal • Stemming • Transform cases A few of these processes have been briefed in the pages ahead.	Depending on the business problem at hand, we can use several of the basic machine learning algorithms to crunch our text to discover patterns. These include : • Word frequencies and word clouds • K-Means cluster analysis • Association rule generators • K nearest neighbors • Neural networks • Naïve Bayes classifiers • N-gram generators	This is an inferential step. The analytics expert has the option to decide if • another refined iteration of the algorithm is justified, or • is the current result good enough to discover unseen patterns among the text

POS Tagging: Parts of speech Tagging

Detailed steps in Text mining

1. **Corpus Creation** - It involves creating a matrix comprising of documents and terms (or tokens). A document can be understood as each row having product description and each column having terms. Terms refers to each word in the description. Usually, the number of documents in the corpus equals to number of rows in the given data.

2. Text Cleaning - It involves cleaning the text in following ways:

Remove words - If the data is extracted using web scraping, you might want to remove html tags.
Remove stop words - Stop words are a set of words which helps in sentence construction and don't have any real information. Words such as a, an, the, they, where etc. are categorized as stop words.
Convert to lower - To maintain a standarization across all text and get rid of case differences and convert the entire text to lower.
Remove punctuation - We remove punctuation since they don't deliver any information.
Remove number - Similarly, we remove numerical figures from text
Remove whitespaces - Then, we remove the used spaces in the text.
Stemming & Lemmatization - Finally, we convert the terms into their root form. For example: Words like playing, played, plays gets converted to the root word 'play'. It helps in capturing the intent of terms precisely.

3. Feature Engineering - To be explained in the next slide

4. Model Building - After the raw data is passed through all the above steps, it become ready for model building. As mentioned above, not all ML algorithms perform well on text data. Naive Bayes is popularly known to deliver high accuracy on text data. In addition, deep neural network models also perform fairly well.

What are Feature Engineering Techniques used in Text Mining ?

Text data offers a wide range of possibilities to generate new features. But sometimes, we end up generating lots of features, to an extent that processing them becomes a painful task. Hence we should meticulously analyze the extracted features.

Below is the list of popular feature engineering methods used:

1. **n-grams** : In the document corpus, 1 word (such as baby, play, drink) is known as 1-gram. Similarly, we can have 2-gram (baby toy, play station, diamond ring), 3-gram etc. The idea behind this technique is to explore the chances that when one or two or more words occurs together gives more information to the model.

2. **TF - IDF** : It is also known as Term Frequency - Inverse Document Frequency. This technique believes that, from a document corpus, a learning algorithm gets more information from the rarely occurring terms than frequently occurring terms. Using a weighted scheme, this technique helps to score the importance of terms. The terms occurring frequently are weighted lower and the terms occurring rarely get weighted higher. * TF is be calculated as: frequency of a term in a document / all the terms in the document. * IDF is calculated as: ratio of log (total documents in the corpus / number of documents with the 'term' in the corpus) * Finally, TF-IDF is calculated as: TF X IDF. Fortunately, R has packages which can do these calculations effort

What are Feature Engineering Techniques used in Text Mining ?

3. **Cosine Similarity** - This measure helps to find similar documents. It's one of the commonly used distance metric used in text analysis. For a given 2 vectors A and B of length n each, cosine similarity can be calculated as a dot product of two unit vectors:

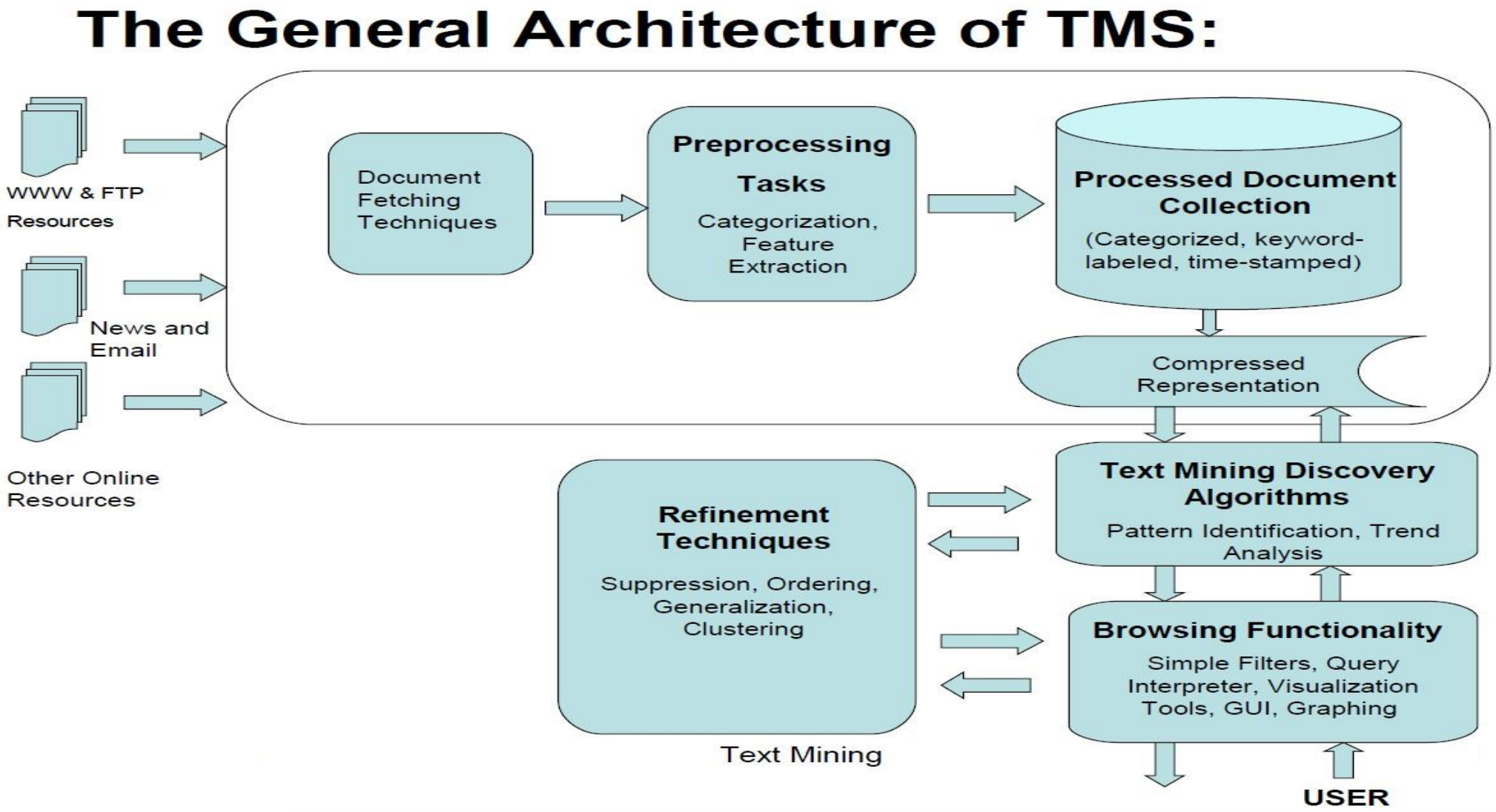
$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

4. **Jaccard Similarity** - This is another distance metric used in text analysis. For a given two vectors (A and B), it can be calculated as ratio of (terms which are available in both vectors / terms which are available in either of the vectors). It's formula is: (A ∩ B)/(A U B). To create features using distance metrics, first we'll create cluster of similar documents and assign a unique label to each document in a new column.

5. **Levenshtein Distance** - We can also use levenshtein distance to create a new feature based on distance between two strings. We won't go into its complicated formula, but understand what it does: it finds the shorter string in longer texts and returns the maximum value as 1 if both the shorter string is found. For example: Calculating levenshtein distance for string "Alps Street 41" and "1st Block, Alps Street 41" will result in 1.

6. **Feature Hashing** - This technique implements the 'hashing trick' which helps in reducing the dimension of document matrix (lesser columns). It doesn't use the actual data, instead it uses the indexes[i,j] of the data, thus it processes data only when needed. And, that's why it takes lesser memory in computation. In addition, there are more techniques which we'll discover while modeling text data in the next section.

Architecture of TM Systems



Software for Text Mining

Software for Text Mining

Product	Preprocess	Associate	Cluster	Summarize	Categorize	API
Commercial						
Clearforest	✓	✓	✓	✓		
Copernic **Summarizer**	✓			✓		
dtSearch	✓	✓		✓		
Insightful **Infact**	✓	✓	✓	✓	✓	✓
Inxight	✓	✓	✓	✓	✓	✓
SPSS **Clementine**	✓	✓	✓	✓	✓	
SAS **Text Miner**	✓	✓	✓	✓	✓	
TEMIS	✓	✓	✓	✓	✓	
WordStat	✓	✓	✓	✓	✓	
Open Source						
GATE	✓	✓	✓	✓	✓	✓
RapidMiner	✓	✓	✓	✓	✓	✓
Weka/KEA	✓	✓	✓	✓	✓	✓
R/**tm**	✓	✓	✓	✓	✓	✓

Text mining packages in R

- Corpora gsubfn kernlab KoNLP
- koRpus `lda lsa maxent
- movMF openNLP qdap RcmdrPlugin.temis
- RKEA RTextTools Rweka skmeans
- Snowball SnowballCtau textcat
- Textir tm tm.plugin.dc tm.plugin.factiva
- tm.plugin.mail topicmodels wordcloud
- Wordnet zipfR

Text mining packages in R

- plyr: Tools for splitting, applying and combining data
- class: Various functions for classification
- tm: A framework for text mining applications
- corpora: Statistics and data sets for corpus frequency data
- snowball: stemmers
- Rweka: interface to Weka, a collection of ML algorithms for data mining tasks
- wordnet: interface to WordNet using the Jawbone Java API to WordNet
- wordcloud: to make cloud of word
- textir: A suite of tools for text and sentiment mining
- tau: Text Analysis Utilities
- topicmodels: an interface to the C code for Latent Dirichlet Allocation (LDA) models and Correlated Topics Models (CTM)
- zipfR: Statistical models for word frequency distributions

Text mining in R

In R, Text mining can be done many ways.

1. Using Text mining function & Regular Expressions
2. Using different packages like tm, NLP etc
3. Using different API's

Structure of Typical Text mining Code

R code snippets to get started with Text Mining (1/2)

```
## Load packages
require(tm)
require(wordcloud)
require(RWeka)

## Read data and create corpus
Data <- read.csv(…)
Corp <- Corpus(DataframeSource(Data), readerControl =
               list(language = "de"))

## Data preprocessing: cleaning and stemming
Corp_proc <- tm_map(Corp, stripWhitespace)
Corp_proc <- tm_map(Corp_proc, tolower)
sw <- c("and",…) # Stop word list
Corp_proc <- tm_map(Corp_proc, removeWords, sw)
Corp_proc <- tm_map(Corp_proc, removePunctuation)
Corp_proc <- tm_map(Corp_proc, removeNumbers)
Corp_stem <- tm_map(Corp_proc, stemDocument,
                    language = "german")
Corp_stem <- tm_map(Corp_stem, stemCompletion, dictionary
                    = Corp_proc, type="prevalent")

## Count frequent single words
dtm1 <- TermDocumentMatrix(Corp_stem)

## Count frequent word pairs
NGram_Tok <- function(x) RWeka::NGramTokenizer(x,
                         Weka_control(min = 2, max = 2))
dtm2 <- TermDocumentMatrix(HCCorpus, control =
                           list(tokenize = NGram_Tok))
```

Load packages: `tm` offers the basic Text Mining framework, `wordcloud` helps to create the same and `RWeka` connects **R** with **Weka**, a collection of machine learning algorithms written in Java that includes text mining procedures

The package `tm` provides `Corpus` as a new data type for collections of texts. Basically a Corpus is a list of texts which can be associated with **meta data** (both, the Corpus and each single text can have meta data). The standard Corpus is **volatile**, i.e. it is fully kept in memory. In case of huge data sets, it is possible to write texts into a **Permanent Corpus** (`PCorpus`) which is stored in a data base on the hard disk.

The **stop word list** contains words that should be removed from all documents as they do not contain valuable information in the analyzed context. It is therefore context related and will be updated gradually.

The `tm_map` procedure takes a function as an argument (e.g. `tolower`) and applies it to each single text of the Corpus subsequently.
Using this function, the data preprocessing steps can be performed, i.e. cleaning, stemming,…

This function computes a **term-document-matrix**: It contains the number of occurrences of each word in each document. The resulting matrix is usually very large (number of cells = number of words * number of documents in Corpus) and sparse (many zeros).

In order to incorporate not only single words but also word pairs, word triples, etc., the so called N Gram **tokenizer** can be applied. In this case, we use a Java implementation from `RWeka`. Also other tokenizers are accessible, for triples (`min = 3` and `max = 3`) or entire phrases (refer to RWeka documentation).

R code snippets to get started with Text Mining (2/2)

```
## Build dummy variables from Term Document Matrix
dummy_dtm1 <- dtm1>0
coef_mat <- matrix(…)
recom <- …

for (k in 1:ncol(dummy_dtm1))
  {
  coef_mat[k,] <- coef(glm(recom ~ dummy_dtm1[,k] + …),
                       family=binomial(link=logit))
  }

## Create wordclouds
wordcloud(Words, Frequencies, …)
```

For modeling the relationship between recommendation and frequently used expressions, a **dummy variable** is created for each phrase indicating if a text contains that phrase.

The `recom` variable contains the recommendation associated with each hotel review as a binary indicator, which is the KPI of interest in this analysis.

Build **predictive models** based on the outcome of the text analysis. In this case, recommendation is modeled dependent on words occurring in each associated text (and other independent variables) by a logistic regression model (one by one).

Draw **word clouds** from most frequent words. This function takes many arguments that help to design a custom layout; colorization of words being one of them.

Text Mining with Regular Expressions

What is a Regular Expression?

Regexes are an extremely flexible tool for finding and replacing **text**. They can easily be applied globally across a document, dataset, or specifically to individual strings.

Example

Data

LastName, FirstName, Address, Phone Baker, Tom,
123 Unit St., 555-452-1324
Smith, Matt, 456 Tardis St., 555-326-4567
Tennant, David, 567 Torchwood Ave., 555-563-8974

Regular Expression to Convert “St.” to “Street”

gsub(“St\\.”, “Street”, data[i])

*Note the double-slash “\\” to escape the ‘.

Benefits of Regex

Flexible (can be applied globally or specifically across data)

Terse (very powerful scripting template)

Portable (sort of) across languages

Rich history

Disadvantages of regex

Non-intuitive

Easy to make errors (unintended consequences)

Difficult to robustly debug

Various flavors may cause portability issues.

Why do this in R?

Easier to locate all code in one place

(Relatively) Robust regex tools

May be the only tool available

Familiarity

Other alternatives?

Perl

Python

Java

Ruby

Others (grep, sed, awk, bash, csh, ksh, etc.)

Components of a Regular Expression

Characters

Metacharacters

Character classes

The R regex functions

Function	Purpose
strsplit()	breaks apart strings at predefined points
grep()	returns a vector of indices where a pattern is matched
grepl()	returns a logical vector (TRUE/FALSE) for each element of the data
sub()	replaces one pattern with another at **first** matching location
gsub()	replaces one pattern with another at **every** matching location
regexpr()	returns an integer vector giving the starting position of the first match, along with a match.length attribute giving the length of the matched text.
gregexpr()	returns an integer vector giving the starting position of the all matches, along with a match.length attribute giving the length of the matched text.

Note: all functions are in the base package

Metacharacter Symbols

Modifier	Meaning
^	anchors expression to beginning of target
$	anchors expression to end of target
.	matches any single character except newline
|	separates alternative patterns
[]	accepts any of the enclosed characters
[^]	accepts any characters but the ones enclosed in brackets
()	groups patterns together for assignment or constraint
*	matches zero or more occurrences of preceding entity
?	matches zero or one occurrences of preceding entity
+	matches one or more occurrences of preceding entity
{n}	matches exactly *n* occurrences of preceding entity
{n,}	matches at least *n* occurrences of preceding entity
{n,m}	matches *n* to *m* occurrences of preceding entity
\	interpret succeeding character as literal

Examples

[A-Za-z]+	**matches one or more alphabetic characters**
.*	**matches zero or more of any character up to the newline**
.*\\.*	**matches zero or more characters followed by a literal .***
(July?)	**Accept 'Jul' or 'July' but not 'Julyy'. Note the space.**
(abc|123)	**Match "abc" or "123"**
[abc|123]	**Match *a, b, c, 1, 2 or 3.The '|' is extraneous.***
^(From|Subject|Date):	**Matches lines starting with "From:" or "Subject:" or "Date:"**

Let's work through some examples...

- LastName, FirstName, Address, Phone Baker,

  Tom, 123 Unit St., 555-452-1324
- Smith, Matt, 456 Tardis St., 555-326-4567
- Tennant, David, 567 Torchwood Ave., 555-563-8974

Locate all phone numbers.

Locate all addresses.

Locate all addresses ending in 'Street' (including abbreviations).

Read in full names, reverse the order and remove the comma.

So how would you write the regular expression to match a calendar date in format “mm/dd/yyyy” or “mm.dd.yyyy”?

Regex to identify date format?

What's wrong with

"[0-9]{2}(.|/)[0-9]{2}(.|/)[0-9]{4}"?

Or with

"[1-12](.|/)[1-31](.|/)[0001-9999]" ?

Dates are not an easy problem because they are not a simple text pattern

Best bet is to validate the textual pattern (mm.dd.yyyy) and then pass to a separate function to validate the date (leap years, odd days in month, etc.)

“^(1[0-2]|0[1-9])(\\.|/)(3[0-1]|[1-2][0-9]|0[1-9])(\\.|/) ([0-9]{4})$”

Supported flavors of regex in R

- POSIX 1003.2

  Perl

Perl is the more robust of the two. POSIX has a few idiosyncracies handling '\' that may trip you up.

Usage Patterns

Data validation

String replace on dataset

String identify in dataset (subset of data)

Pattern arithmetic (how prevalent is string in data?)

Error prevention/detection

Problem Solving & Debugging

Remember that regexes are greedy by default. They will try to grab the largest matching string possible unless constrained.

Dummy data - small datasets

Unit testing - testthis, etc.

Build up regex complexity incrementally

Best Practices for Regex in R

Store regex string as variable to pass to function

Try to make regex expression as exact as possible (avoid lazy matching)

Pick one type of regex syntax and stick with it (POSIX or Perl)

Document all regexes in code with liberal comments use cat() to

verify regex string

Test, test, and test some more

Regex Workflow

Define initial data pattern

Define desired data pattern

Define transformation steps

Incremental iteration to desired regex

Testing & QA

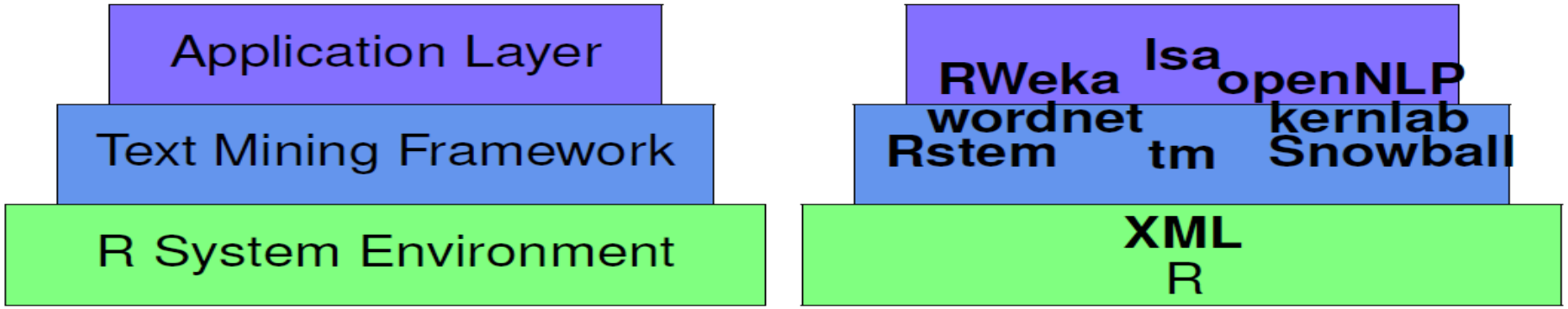
Text Mining with tm package

R package - tm

- ✓ “tm” package for text mining in R.
- ✓ Package offers functionality for managing text documents, performing analysis, and data mining.
- ✓ Numerous extensions and plugins have been developed for handling various formats of input data and producing summary results.
- ✓ It Create a corpus – a collection of text documents
- ✓ It Provide various preprocessing operations
- ✓ It Create a Document-Term/Term-Document matrix
- ✓ It Inspect/manipulate the Document-Term matrix

R package - tm

Input is represented as a Corpus Various readers perform automated pre- processing, feature extraction, and load data into memory.



Conceptual Layers of R framework

Corpus Construction

1. Fetch documents from sources (disk, Internet)
2. Parse document structure (HTML, PDF, `getReaders()`)
3. Extract text and meta information
4. Dynamically create corpus
5. Fill corpus
   - immediately
   - delayed (load on demand)
   - referentially (using pointers to a database)

Create a Corpus

acq

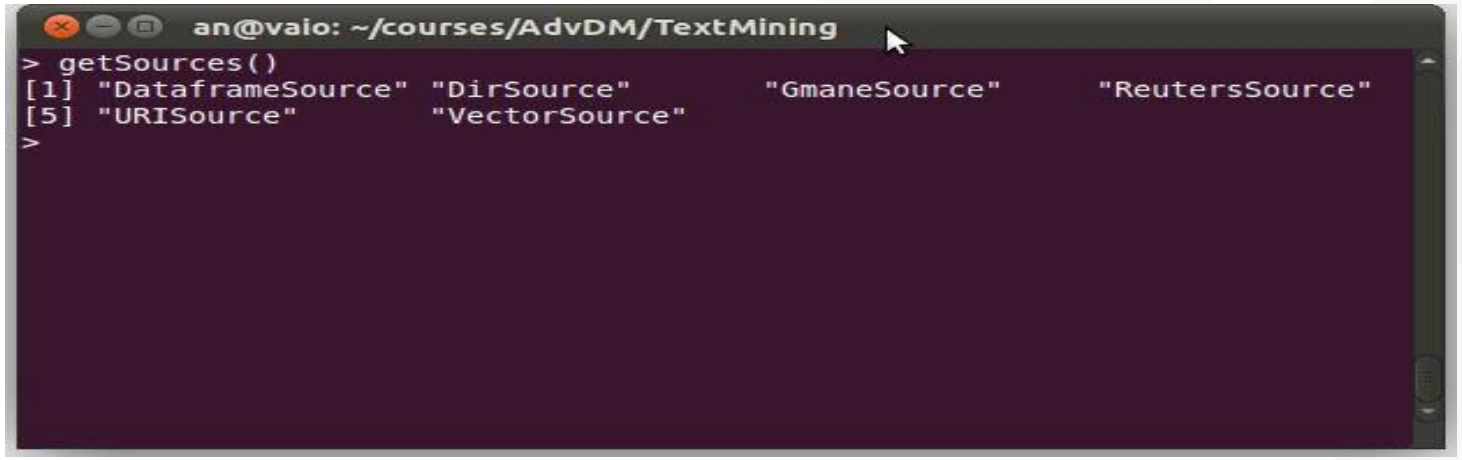
종류:	파일 폴더
위치:	D:₩data₩Reuters
크기:	1.47MB (1,542,498 바이트)
디스크 할당 크기:	123MB (129,826,816 바이트)
내용:	파일 1,981, 폴더 0

```
> (acq <- Corpus(DirSource("d:/data/Reuters/acq")))
A corpus with 1981 text documents
```

Package tm

- ✓ getReaders() gives available readers

- ✓ getSources() gives available sources



Loading data

Loading Data from text documents in a directory

```
> txt <- system.file("texts", "txt", package = "tm")
> (ovid <- Corpus(DirSource(txt),
+                 readerControl = list(language = "lat")))

A corpus with 5 text documents
```

>summary(ovid)
>inspect(ovid[1:2])

Loading data from the web

Can use URISource to create corpus

```
> DosSource <- URISource("http://www.gutenberg.org/files/2554/2554.txt")
> Dostoevsky <- Corpus(DosSource)
> Dostoevsky[[1]][1]
[1] "The Project Gutenberg EBook of Crime and Punishment, by Fyodor Dostoevsky"

> tf <- termFreq(Dostoevsky[[1]])
> head(tf)
      --"that          --and           --he          --she --translator's     --zossimov
           1              2              1              1              1              1
```

Preprocessing

- Original document:

  In 2012, a new mayor of Seoul, PARK Won-Soon, implemented Half-priced tuition, one of his campaign promises as soon as he started his term.

- Preprocessed document:

  mayor seoul park wonsoon implement halfpric tuition campaign promis start term

Preprocessing

- tm_map(Corpus, Function)

- Functions :

  asPlain – removes XML from the document,
  removeSignature – removes the author of the message,
  removeWords, stopwords(language='english') – removes stopwords for the language specified,
  stripWhitespace – removes extra spaces,
  tolower – transforms all upper case letters to lower case,
  removePunctuation – removes punctuation symbols,
  removeNumbers – removes numbers,

- Example of use:

```
> acq <- tm_map(acq, removeNumbers)
```

Processing text

Very easy to process and manipulate text. Stemming a document:

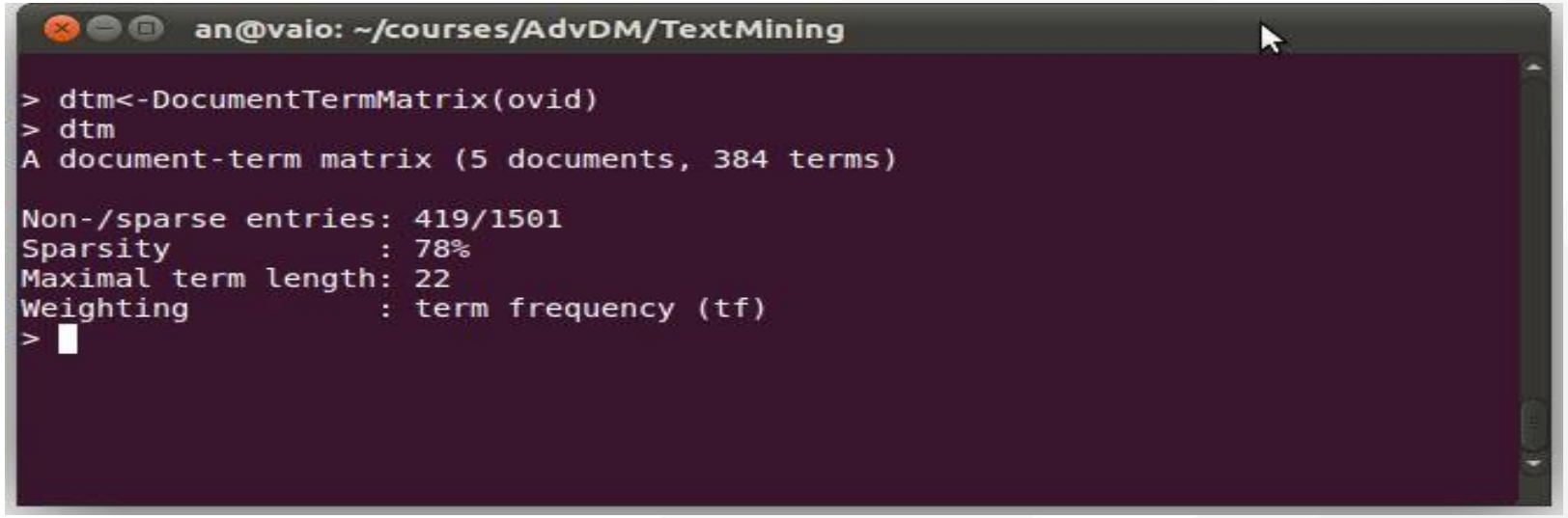
```
> sd <- stemDocument(Dostoevsky[[1]])
> head(sd)
[1] "The Project Gutenberg EBook of Crime and Punishment, by Fyodor Dostoevski"
[2] ""
[3] "This eBook is for the use of anyon anywher at no cost and with"
[4] "almost no restrict whatsoever.  You may copi it, give it away or"
[5] "re-us it under the term of the Project Gutenberg Licens includ"
[6] "with this eBook or onlin at www.gutenberg.org"
```

Unlisting words:

```
> unlist(strsplit(Dostoevsky[[1]][(start+6):(start+8)], split="[[:space:]]", perl=T))
 [1] "On"          "an"          "exceptionally" "hot"
 [5] "evening"     "early"       "in"            "July"
 [9] "a"           "young"       "man"           "came"
[13] "out"         "of"          "the"           "garret"
[17] "in"          "which"       "he"            "lodged"
[21] "in"          "S."          "Place"         "and"
[25] "walked"      "slowly,"     "as"            "though"
[29] "in"          "hesitation," "towards"       "K."
```

Document Term Matrix

Document-Term matrix contains documents as rows and terms as columns:



Document-Term Matrix

IDs 1 : text mining is fun

IDs 2 : a text is a sequence of words

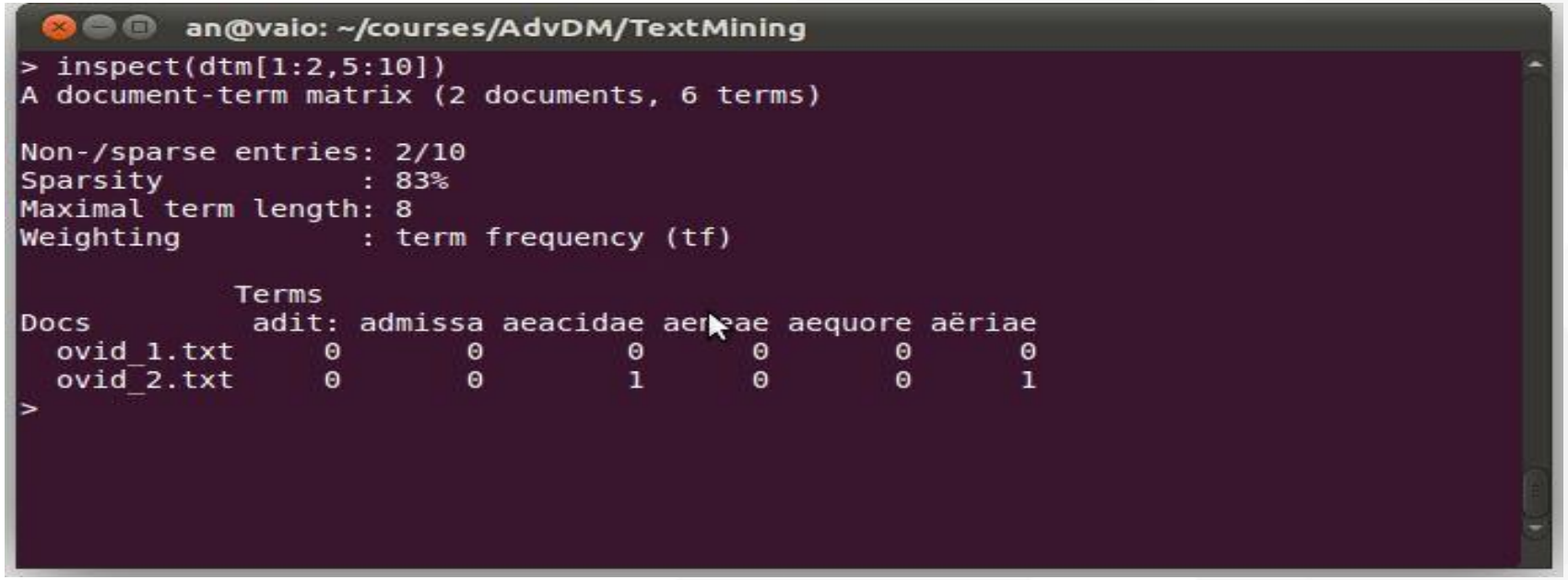
Document-Term matrix

	a	fun	is	mining	of	sequence	text	words
1	0	1	1	1	0	0	1	0
2	2	0	1	0	1	1	1	1

- Creating Term-Document Matrices
  > TermDocumentMatrix(Corpus)
  > DocumentTermMatrix(Corpus)

Document Term Matrix

Can inspect the matrix created:



Term Document Matrix (TDM)

TDM contains terms as rows and documents as columns. Can also create chunks of terms and load.

Feature Selection

- Information gain feature selection

  information.gain() function in package ***FSelector***

- X^2-test feature selection

  chi.squared() function in package ***FSelector***

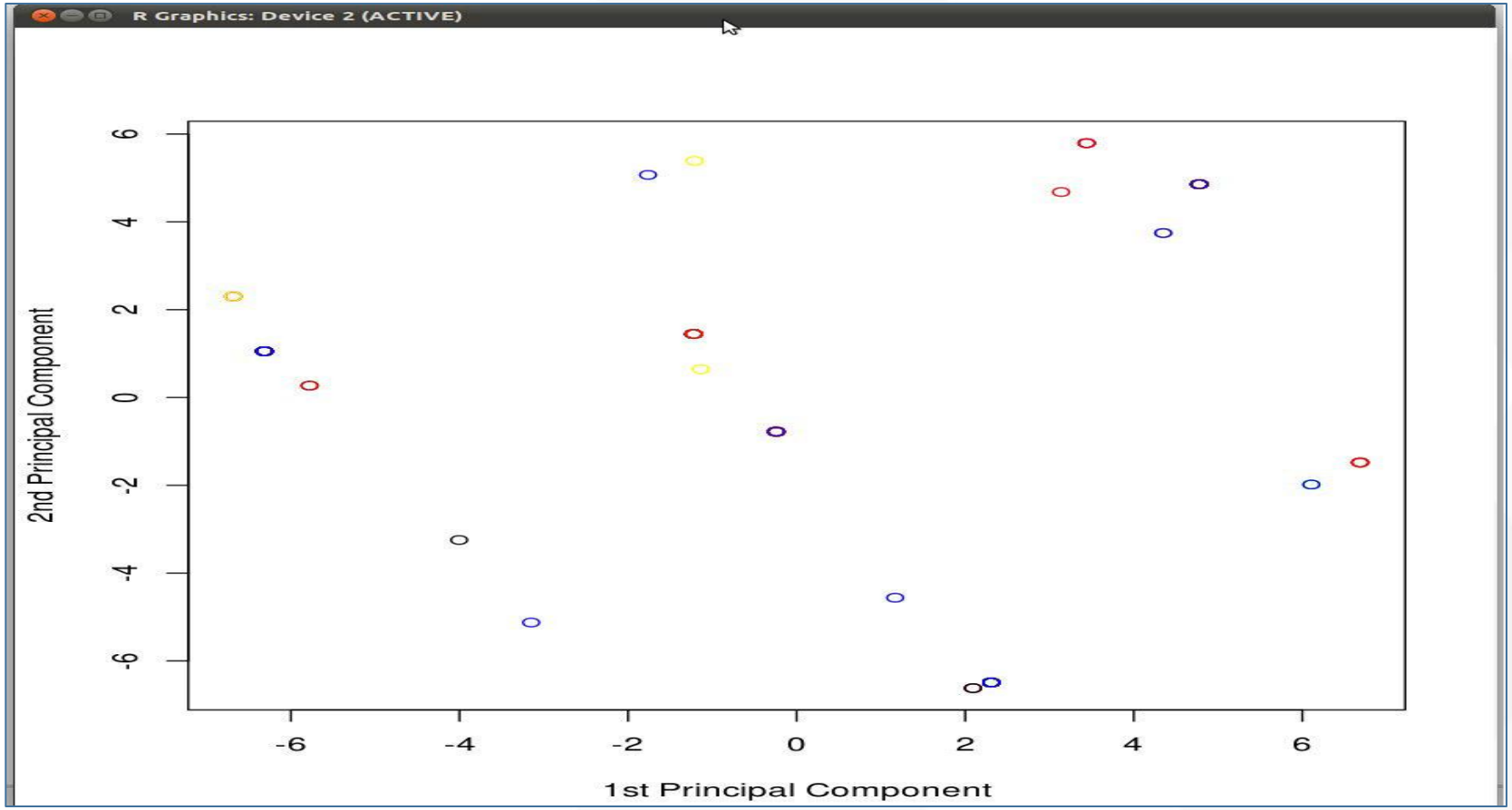
- Example of use:

```
> weights <- information.gain(type~., reutData)
> sel.feature <- cutoff.k(weights, 5000)
> reutData <- reutData[,sel.feature]
```

PCA Analysis using TDM

TDM matrix can be used for PCA analysis and plotting

PCA plot using 2 features



Simple Text Clustering

- Hierarchical clustering
  hclust() function in package ***stats***
  agnes() function in package ***cluster***

- k-means clustering
  kmeans() function in package ***stats***

- Model-based clustering
  mclust() function in package ***mclust***

Simple Text Classification

- k-nearest neighbor classification
  knn() function in package ***class***

- Support vector machine classification
  ksvm() function in package ***kernlab***
  svm() function in package ***el071***
  svmlight() function in package ***klaR***

- Naive Bayes classification
  naiveBayes() function in package ***el071***

Using tm Plugins

- ✓ The tm package has become very popular in a number of industries – academics, bioinformatics, linguistics, finance, actuary, news analysis, etc.
- ✓ Number of people have developed plugin for tm package.

eg: tm.plugin.webmining, tm.plugin.sentiment, tm.plugin.dc (for distributed computing)

Use Cases & Case Studies

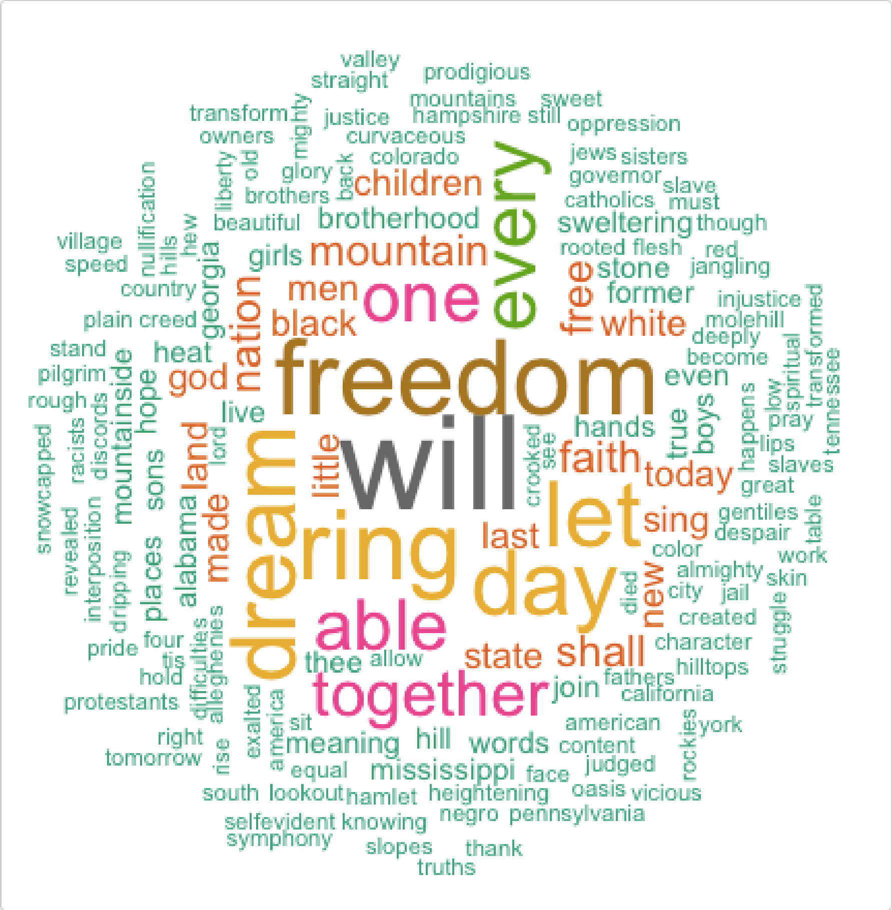
Use Cases & Exercises

1. **Word cloud analysis – Frequent word analysis - Text segmentation (Topics Modelling)**
2. **Speech Classification**
3. **Survey Sentiment Analysis**
4. **Speech Classification**
5. **News Aggregation**
6. **Designing Search Engine**
7. **Natural Language Processing**
8. **Clustering Uncategorized RSS Feeds**
9. **Social Media analysis with Face book & Twitter data**
10. **Predicting Gender of Blog Writer using Classification**

Sample analysis

Word clouds

- A **word cloud** is a **text mining** method that allows us to highlight the most frequently used keywords in a paragraph of texts.
- It is also referred to as a **text cloud** or **tag cloud**.
- A **text mining** package (tm) and **word cloud generator** package (**wordcloud**) are available in R for helping us to analyze texts and to quickly visualize the keywords words as a **word cloud**.



Word clouds

3 reasons you should use word clouds to present your text data

- **Tag cloud** is a powerful method for **text mining** and, it add simplicity and clarity. The most used keywords stand out better in a word cloud
- **Word clouds** are a potent communication tool. They are easy to understand, to be shared and are impactful
- **Word clouds** are visually engaging than a table data

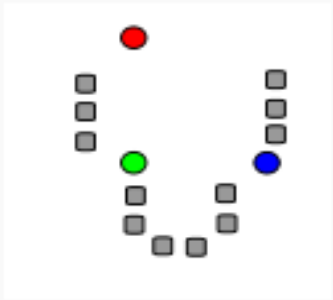
Who is using word clouds ?

- Researchers : for reporting qualitative data
- Marketers : for highlighting the needs and pain points of customers
- Educators : to support essential issues
- Politicians and journalists
- social media sites : To collect, analyze and share user sentiments

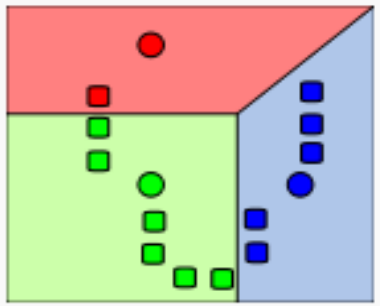
K-Means clustering

- unsupervised learning
- group **n** documents into **k** clusters

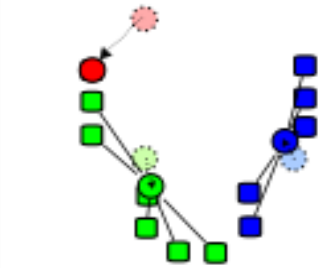
Demonstration of the standard algorithm



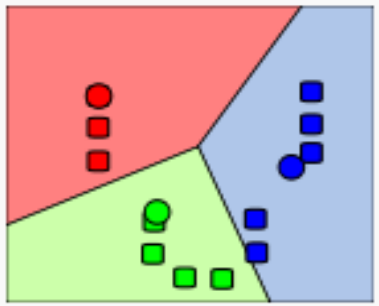
1) *k* initial "means" (in this case *k*=3) are randomly selected from the data set (shown in color).



2) *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.



3) The centroid of each of the *k* clusters becomes the new means.



4) Steps 2 and 3 are repeated until convergence has been reached.

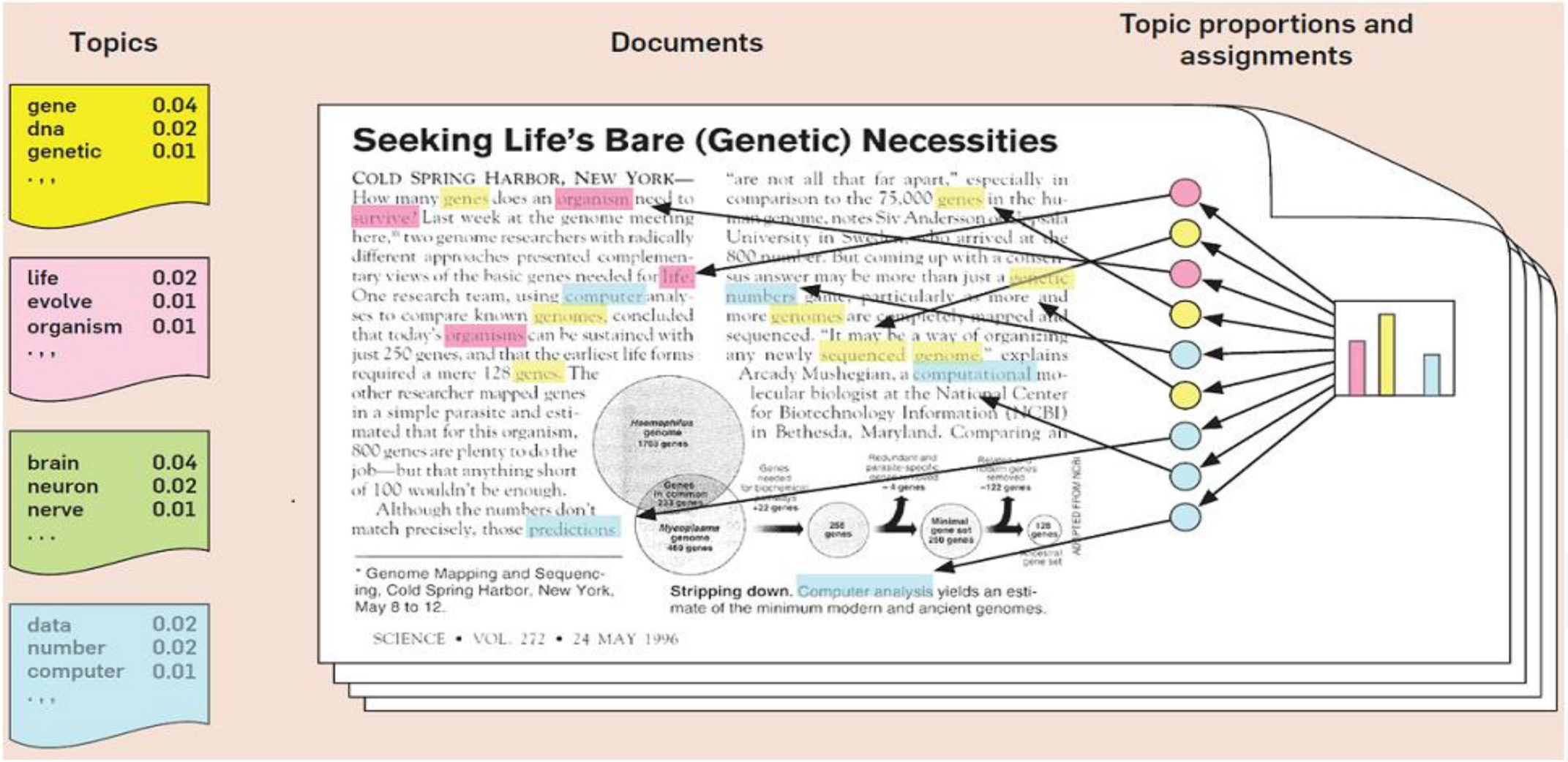
Weighting terms for clustering

- term frequency-inverse document frequency (tf-idf)
- offset by term frequency in the corpus

Example: N = # documents, d = # documents with term

- "information content" of a term: log(N/d)
  - rare term = high idf: log(100/4) = 4.64
  - common term = low idf: log(100/60) = 0.74

Topic modeling with topicmodels



Blei, 2012, Communications of the ACM

Example: Speech Classification

KNN for speech classification

Datasets:

- Size: 40 instances
- Barak Obama 20 speeches
- Mitt Romney 20 speeches

- Training datasets: 70% (28)
- Test datasets: 30% (12)

- **Accuracy:** on average more than 90%

Speech Classification Implementation in R

- **#initialize the R environment**

```
libs<-c("tm","plyr","class")
lapply(libs,require,character.only=TRUE)
```

- **#Set parameters / source directory**

```
dir.names<-c("obama","romney")
path<-"E:/Ashraf/speeches"
```

- **#clean text / preprocessing**

```
cleanCorpus<-function(corpus){
 corpus.tmp<-tm_map(corpus,removePunctuation)
 corpus.tmp<-tm_map(corpus.tmp,stripWhitespace)
 corpus.tmp<-tm_map(corpus.tmp,tolower)
 corpus.tmp<-tm_map(corpus.tmp,removeWords,stopwords("english"))
 return (corpus.tmp)
}
```

Speech Classification Implementation in R

- **#build term document matrix**

```
generateTDM<-function(dir.name,dir.path){
 s.dir<-sprintf("%s/%s",dir.path,dir.name)
 s.cor<-Corpus(DirSource(directory=s.dir,encoding="ANSI"))
 s.cor.cl<-cleanCorpus(s.cor)
 s.tdm<-TermDocumentMatrix(s.cor.cl)
 s.tdm<-removeSparseTerms(s.tdm,0.7)
 result<-list(name=dir.name,tdm=s.tdm)
}

tdm<-lapply(dir.names,generateTDM,dir.path=path)
```

Speech Classification Implementation in R

- **#attach candidate name to each row of TDM**

```
bindCandidateToTDM<-function(tdm){
 s.mat<-t(data.matrix(tdm[["tdm"]]))
 s.df<-as.data.frame(s.mat,StringAsFactors=FALSE)
 s.df<-cbind(s.df,rep(tdm[["name"]],nrow(s.df)))
 colnames(s.df)[ncol(s.df)]<-"targetcandidate"
 return (s.df)
}

candTDM<-lapply(tdm,bindCandidateToTDM)
```

Speech Classification Implementation in R

- **#stack the TDMs together (for both Obama and Romnie)**

```
tdm.stack<-do.call(rbind.fill,candTDM)
tdm.stack[is.na(tdm.stack)]<-0
```

- **#hold-out / splitting training and test data sets**

```
train.idx<-sample(nrow(tdm.stack),ceiling(nrow(tdm.stack)*0.7))
test.idx<-(1:nrow(tdm.stack))[-train.idx])
```

Speech Classification Implementation in R

- **#model KNN**

```
tdm.cand<-tdm.stack[,"targetcandidate"]
tdm.stack.nl<-tdm.stack[,!colnames(tdm.stack)%in%"targetcandidate"]

knn.pred<-knn(tdm.stack.nl[train.idx,],tdm.stack.nl[test.idx,],tdm.cand[train.idx])
```

- **#accuracy of the prediction**

```
conf.mat<-table('Predictions'=knn.pred,Actual=tdm.cand[test.idx])
(accuracy<-(sum(diag(conf.mat))/length(test.idx))*100)
```

- **#show result**

```
show(conf.mat)
show(accuracy)
```

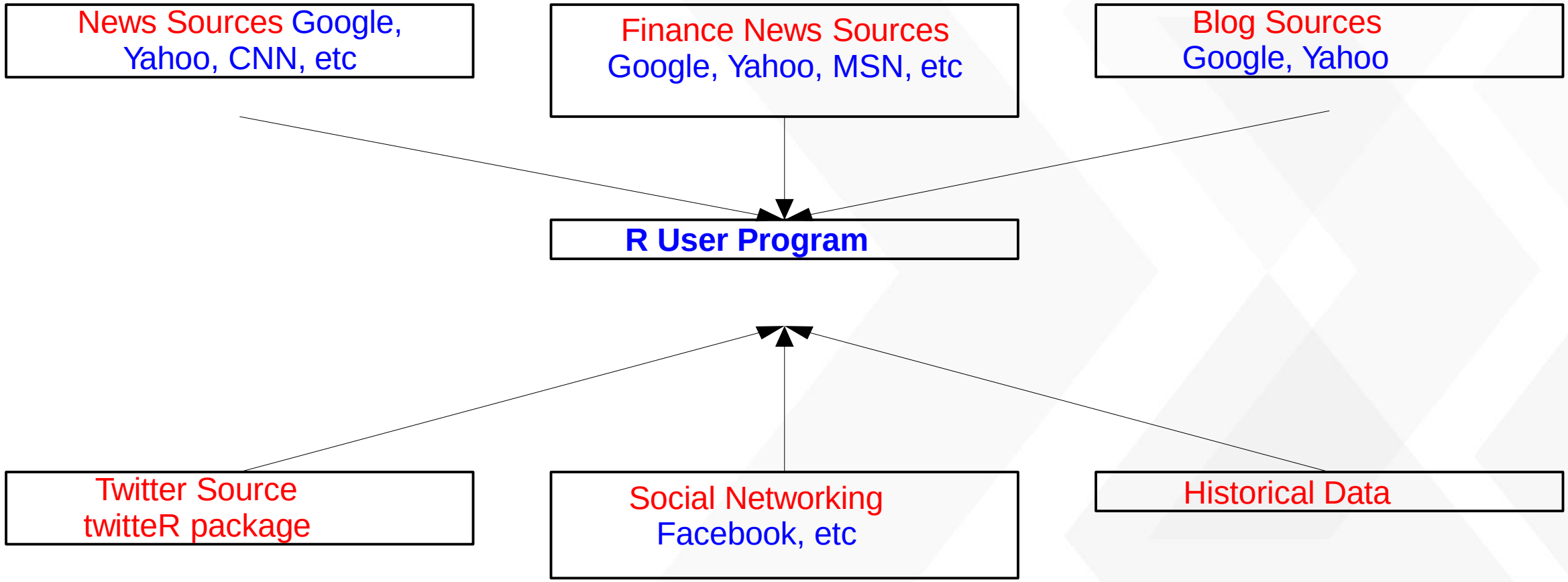
Example: Survey sentiment analysis

Example: Survey sentiment analysis and clustering

Objective	A major US retail bank conducted a diagnostic around the workplace technologies that they are using through a collection of surveys from an internal employee satisfaction survey. The task was to find out the themes on workplace technologies (e.g. lotus notes, video conferencing, Wi-Fi, OS etc.) and the effect of the technology on productivity.
Analysis Input	An excel file showing the results of the survey along with the comments
Analysis Process	▪ The input of the text file was given directly to the tropes software. Tropes has the option of customizing and adding our own scenarios, but for this particular case, it is not required. ▪ The process included - sentence and proposition Hashing, ambiguity solving (with respect to the words of the text), detection of episodes, detection of the most characteristic parts of text, layout and display of the result. ▪ During the process, the software will: — assign all the significant words to the above categories — analyze their distribution into subcategories (Word categories, Equivalent classes, see below) — examine their occurrence order, both within the propositions (Relations, Actant and Acted) throughout the text
Analysis Output	Through the use of word counts for various relations and scenarios, a table was compiled showing the themes in the technology environment

Text Analytics Example: News Aggregation

How to gather news using R



Read News using R

Let's find out what's going on at SMU.

```
> corpus<-WebCorpus(GoogleNewsSource("Southern Methodist University"))
> corpus[[1]]
SMU's Engaged Learning Day inspires student projects
Email: brekow@smu.edu
Published: Monday, February 13, 2012
```

Headlines:

```
an@vaio: ~/courses/AdvDM/TextMining

> sapply(corpus, function(x) {attr(x, "Heading")})
  [1] "SMU's Engaged Learning Day inspires student projects - The Daily Campus"

  [2] "Journalism graduate nominated for four Emmy awards - The Daily Campus"

  [3] "Senate candidate can't escape sports scandals - The Associated Press"

  [4] "SMU sends 18 to Midwest LGBT conference - Dallas Voice"

  [5] "Baylor School of Music Lyceum Series Welcomes Art Historian for Lecture -
 Baylor University"
  [6] "Brooke Reyes receives bachelor's degree from Southern Methodist Universit
y - Your Houston News"
  [7] "Review: New York Baroque Dance Company | Dallas Bach Society - TheaterJon
es Performing Arts News in North Texas"
  [8] "College notes: 02.13.12 - Corpus Christi Caller Times"

  [9] "In Albania, Can a US Diploma Deliver? - New York Times"

 [10] "Closure Takes Top Spot at 2012 Indie Game Challenge - MarketWatch (press
release)"
 [11] "Senate candidate can't escape sports scandals - Houston Chronicle"
```

Financial News

GoogleFinanceSource can be used to get latest news about any listed company

Example:

- corpus <- WebCorpus(GoogleFinanceSource("NASDAQ:MSFT")) Retrives news stories from Google about Microsoft Corporation and creates a corpus.

corpus <- WebCorpus(YahooFinanceSource("MSFT"))

Retrives news stories from Yahoo about Microsoft Corporation and creates a corpus.

corpus <- WebCorpus(TwitterSource("Microsoft"))

Retrives news stories from Twitter about Microsoft Corporation and creates a corpus.

Financial Data

R package quantmod can be used to obtain latest stock market data.

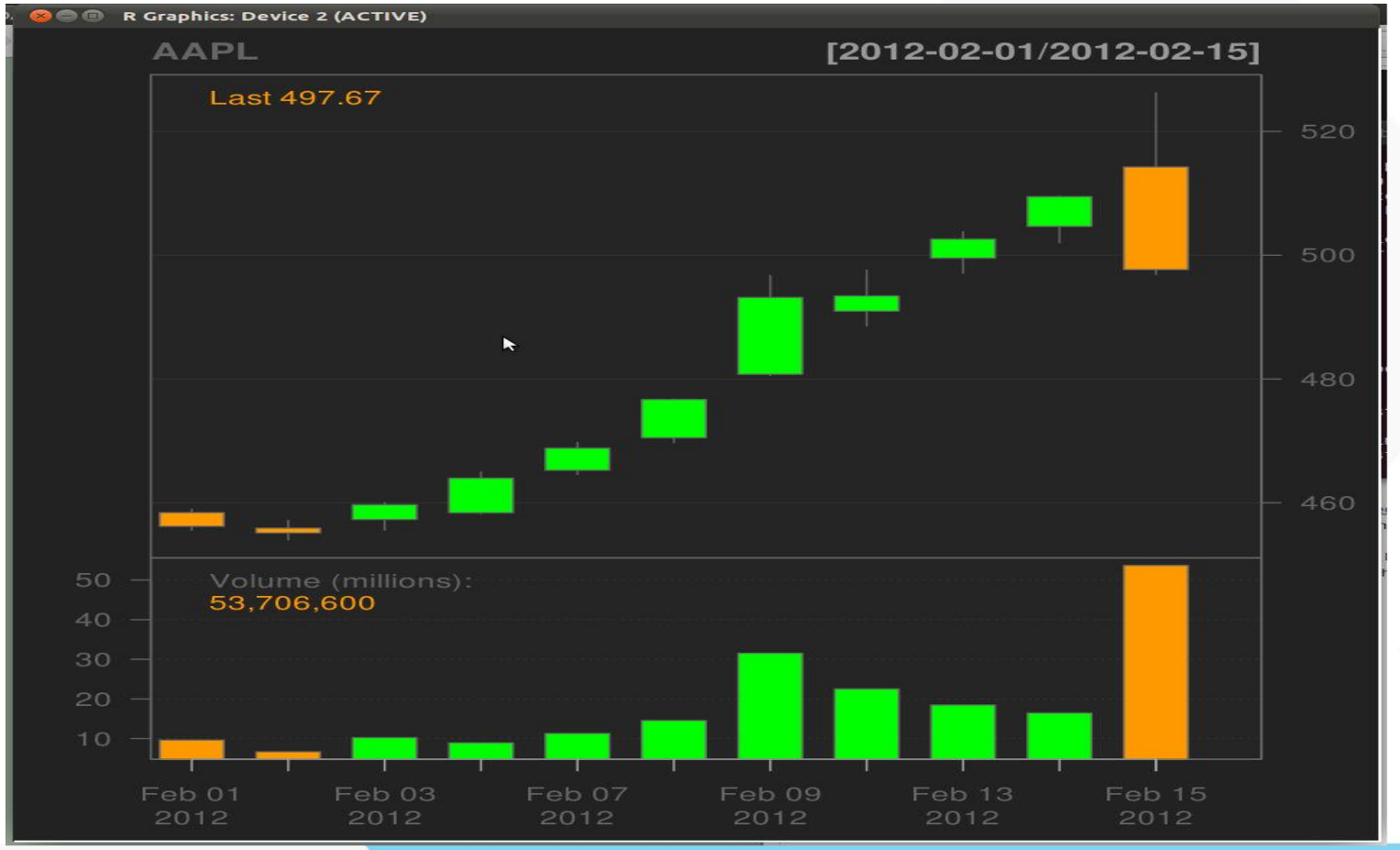
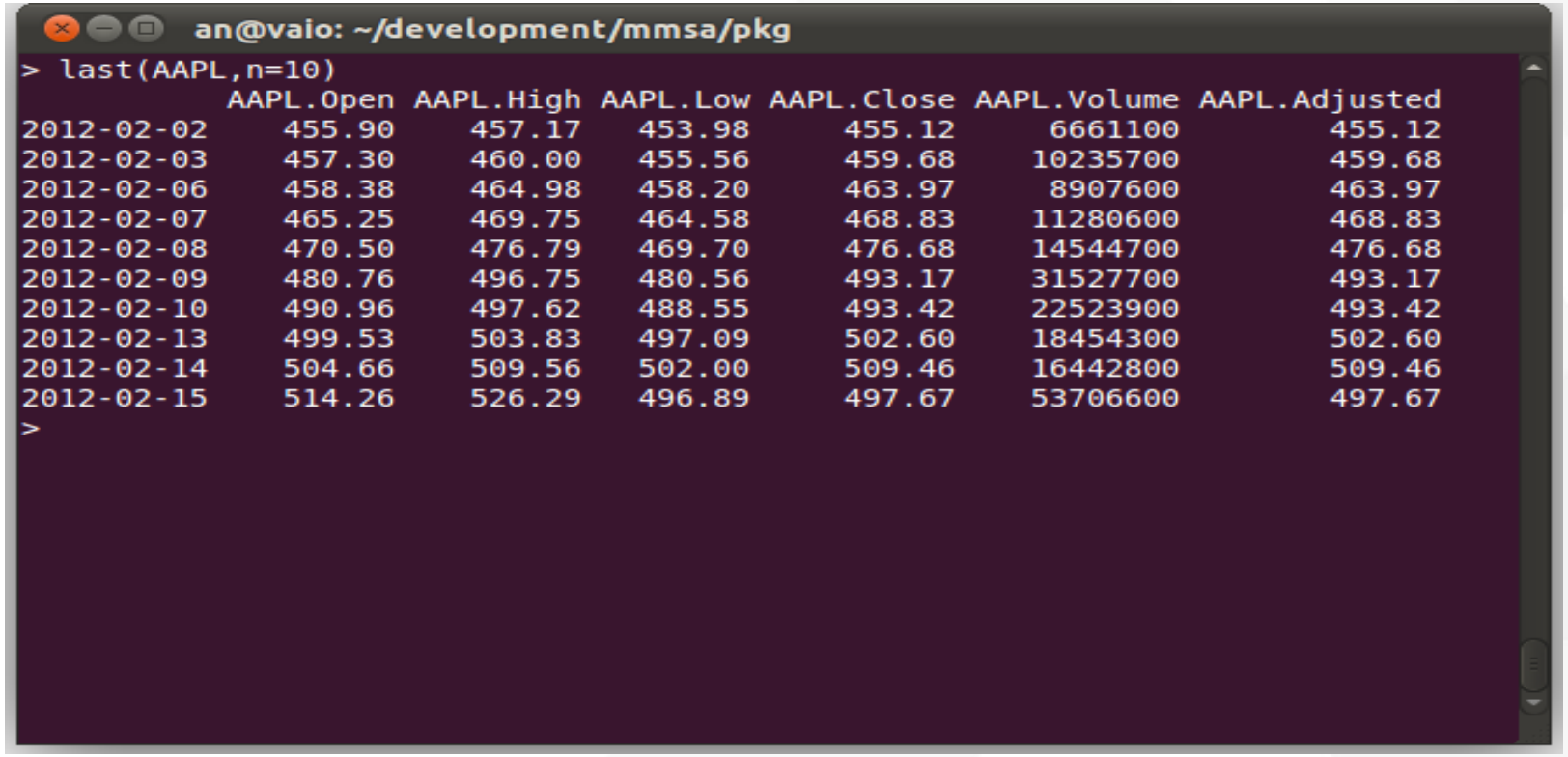


Chart of Apple (NASDAQ:AAPL) for month of February

Financial Data

Can also download data as a matrix



Financial News

Let's get news about Apple Corp.

```
> corpusAAPL <- WebCorpus(GoogleFinanceSource("NASDAQ:AAPL"))
> sapply(corpusAAPL, function(x) {attr(x, "Heading")})
 [1] "Apple&#39;s latest PC OS gets even more iOS-like"

 [2] "OS X 10.8 Mountain Lion Growls at the Masses"

 [3] "Apple Reverses, Stocks Top Out"

 [4] "Dominant Apple Looks To Cripple Rival Samsung"

 [5] "Amazon Declines on Morgan Stanley Downgrade"

 [6] "Apple&#39;s 4Q Global Tablet Market Share Falls To 57% From 64% In 3Q"

 [7] "Apple responds to furor over info-stealing apps"

 [8] "Apple Share Run Paused"
```

How to get News Sentiment

How can we **automatically** analyze news and get a feel whether it conveys positive or negative sentiments.

News Sentiment

R package tm.plugin.tags can help us.

Contains large listing of positive and negative terms that can be used to tag news items.

```
> require("tm.plugin.tags")
> control <- list(stemming = TRUE)
> sample(tm_get_tags("Negativ", control = control), 10)
 [1] "gloomi"      "muddi"       "betray"      "disprov"     "substitut"
 [6] "unnecessari" "cannib"      "hazi"        "undon"       "burn"

> sample(tm_get_tags("Positiv", control = control), 10)
 [1] "humbl"       "fortun"      "spectacular" "respons"     "promin"
 [6] "hilari"      "glad"        "versatil"    "pleasant"    "golden"
```

News Tagging

Let's see which terms are tagged as positive in news for Apple Corp.

```
> colnames(AAPL_dtm_reduced)[which(which_pos==TRUE)]
  [1] "accept"     "accord"     "adjust"     "admit"      "agreement"
  [6] "aid"        "allow"      "appeal"     "approach"   "asset"
 [11] "attract"    "basic"      "benefit"    "board"      "bolster"
 [16] "boost"      "call"       "common"     "confer"     "consent"
 [21] "contact"    "content"    "correct"    "credit"     "deal"
 [26] "discuss"    "entertain"  "enthusiasm" "establish"  "excel"
 [31] "fair"       "familiar"   "favor"      "forward"    "free"
 [36] "fresh"      "friend"     "gain"       "ame"        "glow"
 [41] "gold"       "grace"      "hand"       "haven"      "health"
 [46] "help"       "hit"        "home"       "hope"       "hug"
 [51] "impress"    "inform"     "joke"       "keen"       "kid"
 [56] "law"        "lead"       "legal"      "live"       "loyal"
 [61] "main"       "major"      "matter"     "meet"       "offer"
 [66] "offset"     "partner"    "pass"       "patient"    "permit"
 [71] "plain"      "popular"    "premium"    "prime"      "pro"
 [76] "profit"     "progress"   "protect"    "real"       "regard"
 [81] "respect"    "return"     "rich"       "robust"     "round"
 [86] "safe"       "serious"    "share"      "smile"      "smitten"
 [91] "sought"     "special"    "stand"      "straight"   "success"
 [96] "suit"       "support"    "talent"     "thank"      "tradition"
[101] "travel"     "true"       "truth"      "understand" "uphold"
```

Text Analytics Information Retrieval – Search Engine

Search Engine

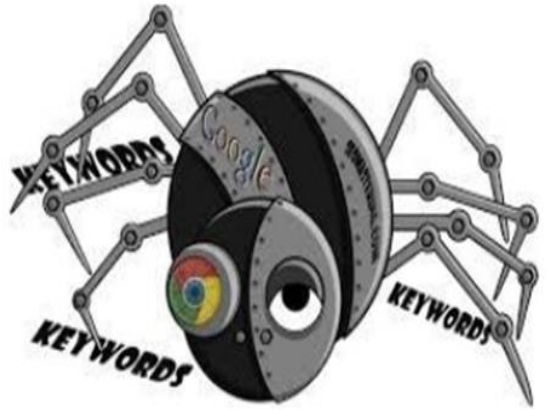
- Web based search engines are the drivers of the 21st century information infrastructure. The tech giants of our life time have built entire business models worth billions of dollars off of a fundamentally simple concept:

- How can I find the information I am looking for on the internet?

- Of course, there is more to this idea when we speak in business applications. How is the ranking determined? If someone, pays google for a higher ranking, how is this incorporate into the results, etc…

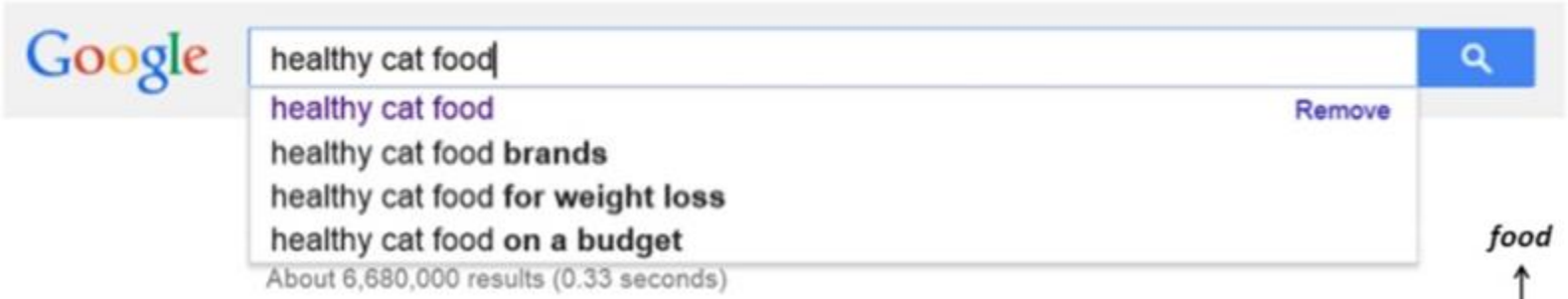
Search Engine

- Search engines are a practical application of text analytics which bridges the gap between unstructured and structured data analytics.
- The basic operating principle is that the search engine provider categorizes the websites (documents) of interest and indexes them using some criterion. We then specify our search parameter and pass this through the search query engine which determines a ranking of results.
- The results with the highest ranking will be the best match based upon our search algorithm.
- For our example, we're going to use a tried and true method for our search algorithm, which origins are from the 1960's. We're going to implement the vector space model of information retrieval in R.

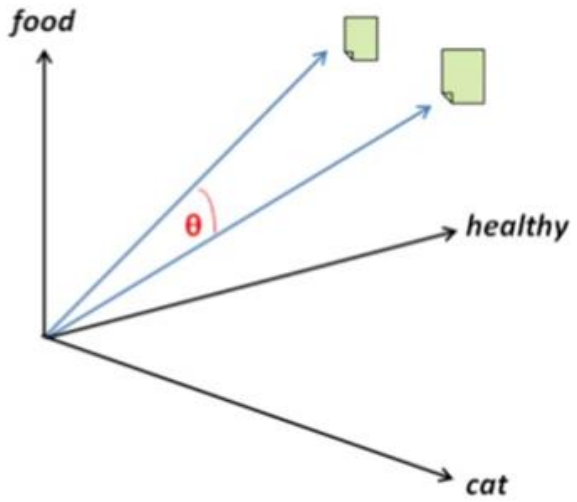


Search Query

- We will build our search engine to find from a group of 7 websites (text documents) the best ranking in descending order.
- We will use the search criteria " healthy cat food" as the query for the analysis.



- A visualization of the vector based space information retrieval model.



Build Corpus

We need to first construct a corpus (a collection of texts) using the 7 various websites (documents).

Here is the example of the unstructured text that has been indexed to apply the query results against.

Web Page	Text Field
1	"Stray cats are running all over the place. I see 10 a day!"
2	"Cats are killers. They kill billions of animals a year."
3	"The best food in Columbus, OH is the North Market."
4	"Brand A is the best tasting cat food around. Your cat will love it."
5	"Buy Brand C cat food for your cat. Brand C makes healthy and happy cats."
6	"The Arnold Classic came to town this weekend. It reminds us to be healthy."
7	"I have nothing to say. In summary, I have told you nothing."

Most of the documents contain some reference to cats, healthy, or food with the exception of document #7.

For simplicity sake, we are going to also include the search query "Healthy Cat Food" into the same corpus.

Preparing the Corpus for Analysis

- In order to improve the quality of our search engines results, we will need to first prepare the text data for further analysis.

- This process consists of the following steps:

  - Remove punctuation
  - Lemmatization or stemming of words (root form)
  - Shift terms to lower case
  - Remove any numbers from the text
  - Strip off any unnecessary white space

Preparing the Corpus for Analysis

- Lets take a look at the following text from our search engine.

Stray cats are running all over the place. I see 10 a day!

- Now lets remove the punctuation.

Stray cats are running all over the place I see 10 a day

- Stem terms to the root form.

Stray cat are run all over the place I see 10 a day

Preparing the Corpus for Analysis

- Remove any numbers.

Stray cat are run all over the place I see a day

- Adjust terms to lower case.

stray cat are run all over the place i see a day

- Remove any additional white space.

stray cat are run all over the place i see a day

Create a Term Document Matrix

Term Document Matrix	
A term-document matrix (14 terms, 8 documents)	
Non-/sparse entries	21/91
Sparsity	: 81%
Maximal term length	: 8
Weighting	: term frequency (tf)

Terms	Web Page 1	Web Page 2	Web Page 3	Web Page 4	Web Page 5	Web Page 6	Web Page 7	Query
all	1	0	0	0	0	0	0	0
and	0	0	0	0	1	0	0	0
anim	0	1	0	0	0	0	0	0
are	1	1	0	0	0	0	0	0
arnold	0	0	0	0	0	1	0	0
around	0	0	0	1	0	0	0	0
best	0	0	1	1	0	0	0	0
billion	0	1	0	0	0	0	0	0
brand	0	0	0	1	2	0	0	0
buy	0	0	0	0	1	0	0	0
came	0	0	0	0	0	1	0	0
cat	1	1	0	2	3	0	0	1
classic	0	0	0	0	0	1	0	0
columbus	0	0	1	0	0	0	0	0
classic	0	0	0	0	0	1	0	0
columbus	0	0	1	0	0	0	0	0

This row contains values from the query parameters as well.

Term Document Weights

- The values of in our document matrix are simple term frequencies.
- This is fine, but other heuristics are available. For instance, rather than a linear increase in the term frequency, tf , perhaps sqrt(tf) or log(tf) would provide a more reasonable diminishing returns on word counts within documents.
- Rare words can also get a boost. The word "healthy" appears in only one document, whereas "cat" appears in four. A word's document frequency, df, is the number of documents that contain it, and a natural choice is to weight words inversely proportional to their df's.
- As with term frequency, we may use logarithms or other transformations to achieve the desired effect.
- Different weighting choices are often made for the query and the documents.

Term Document Weights

- For both the document and the query, we choose the following weights:

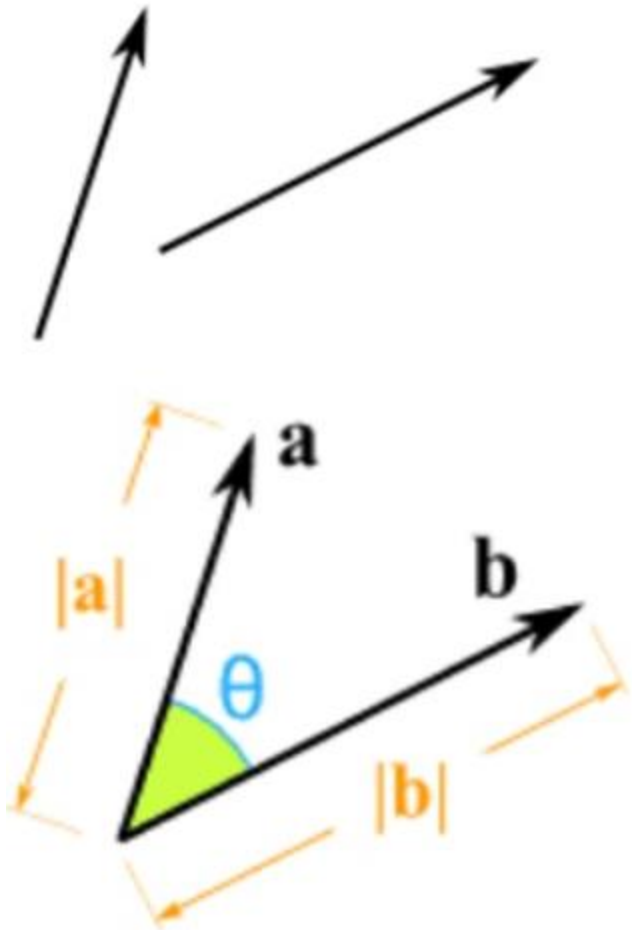
If tf = 0, then 0, otherwise (1 + Log2(tf)) * Log2(N / df))

- We implement this weighting function across entire rows of the term document matrix, and therefore our weighting function must take a term frequency vector and a document frequency scalar as inputs.

Terms	Web Page 1	Web Page 2	Web Page 3	Web Page 4	Web Page 5	Web Page 6	Web Page 7	Query
cat	1	1	0	2	3	0	0	1

Terms	Weighting 1	Weighting 2	Weighting 3	Weighting 4	Weighting 5	Weighting 6	Weighting 7	Query
cat	0.8073549	0.8073549	0	1.61471	2.086982	0	0	0.807355

Dot product Geometry



A benefit of being in the vector space is the use of its dot product or scalar product.

- For vectors a and b, the geometric definition of the dot product is:

$$a \cdot b = ||a||\,||b|| cos\Theta$$

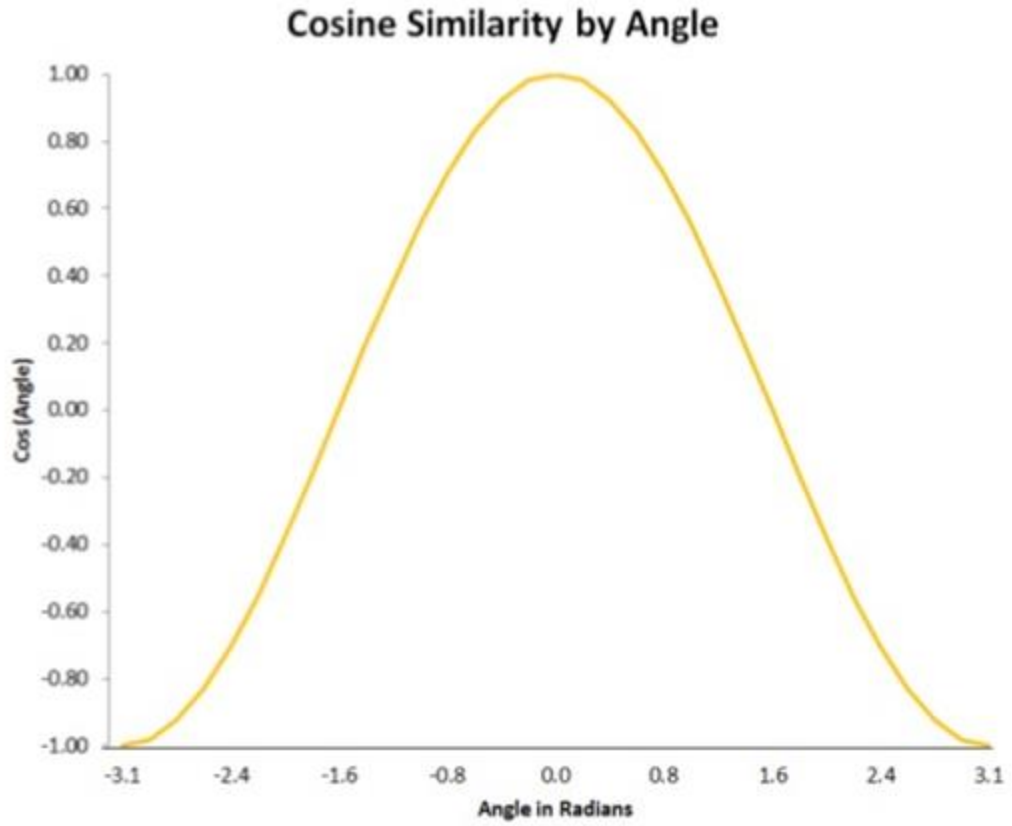
- where $\cdot$ is the Euclidean norm (the root sum of squares) and Θ is the angle between a and b.

Further Normalization

In fact, we can work directly with the cosine of Θ.

- For theta in the interval [-π, - π] , the endpoints are orthogonally (totally unrelated documents) and the center, zero, is complete collinear (maximally similar documents).
- We can see that the cosine decreases from its maximum value of 1.0 as the angle departs from zero in either direction.

- We may furthermore normalize each column vector in our matrix so that its norm is one.
- Now the dot product is cos Θ.



Terms	Weighting 1	Weighting 2	Weighting 3	Weighting 4	Weighting 5	Weighting 6	Weighting 7	Query
cat	0.1044566	0.1128249	0	0.2378746	0.22591472	0	0	0.347026

Matrix Multiplication

- Keeping the query alongside the other documents let us avoid repeating the same steps.
- But now it's time to pretend it was never there.

```
query.vector <- tfidf.matrix[, (N.docs + 1)]
tfidf.matrix <- tfidf.matrix[, 1:N.docs]
```

- With the query vector and the set of document vectors in hand, it is time to go after the cosine similarities. These are simple dot products as our vectors have been normalized to unit length.

- Recall that matrix multiplication is really just a sequence of vector dot products. The matrix operation below returns values of cosine Θ for each document vector and the query vector.

```
doc.scores <- t(query.vector) %*% tfidf.matrix
```

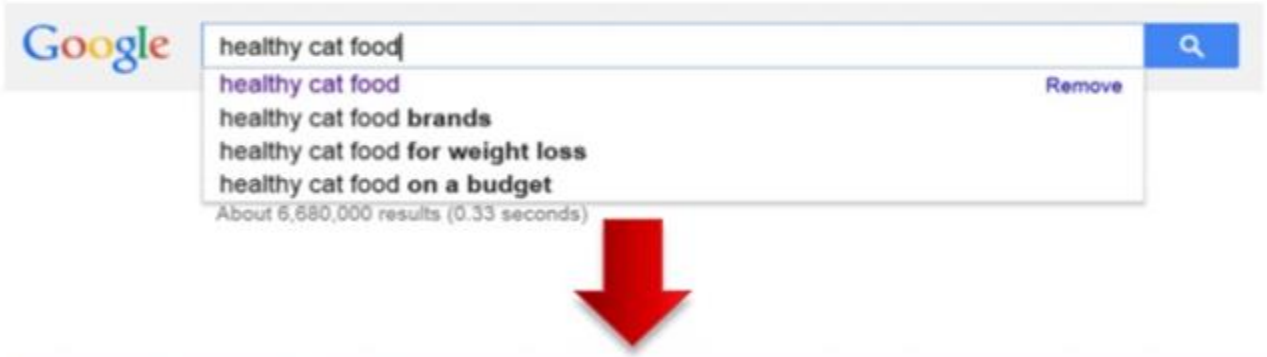
Matrix Multiplication

- With scores in hand, rank the documents by their cosine similarities with the query vector.

```
results.df <- data.frame(doc = names(doc.list), score = t(doc.scores),
                         text = unlist(doc.list))
results.df <- results.df[order(results.df$score, decreasing = TRUE), ]
```

$$\underbrace{\begin{bmatrix}1 & 2 & 3\end{bmatrix}}_{1\times 3} \cdot \underbrace{\begin{bmatrix}2 & 1 & 3\\ 3 & 3 & 2\\ 4 & 1 & 2\end{bmatrix}}_{3\times 3} = \begin{bmatrix}1\cdot 2+2\cdot 3+3\cdot 4\\ 1\cdot 1+2\cdot 3+3\cdot 1\\ 1\cdot 3+2\cdot 2+3\cdot 2\end{bmatrix} = \underbrace{\begin{bmatrix}20\\ 10\\ 13\end{bmatrix}}_{1\times 3}$$

Search Engine Results



Web Page	Score	Text Field
5	0.344	Buy Brand C cat food for your cat. Brand C makes healthy and happy cats.
6	0.183	The Arnold Classic came to town this weekend. It reminds us to be healthy.
4	0.177	Brand A is the best tasting cat food around. Your cat will love it.
3	0.115	The best food in Columbus, OH is the North Market.
2	0.039	Cats are killers. They kill billions of animals a year.
1	0.036	Stray cats are running all over the place. I see 10 a day!
7	0.000	I have nothing to say. In summary, I have told you nothing.

- Our "best" document, at least in an intuitive sense, comes out ahead with a score nearly twice as high as its nearest competitor.
- Notice however that this next competitor has nothing to do with cats.
- This is due to the relative rareness of the word "healthy" in the documents and our choice to incorporate the inverse document frequency weighting for both documents and query.
- Fortunately, the profoundly uninformative document 7 has been ranked dead last.

Text Analytics Example: Natural Language Processing

Natural Language Processing

Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages.

As such, NLP is related to the area of human–computer interaction.

Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation.

Natural Language Processing in R

- There is an R package called openNLP which contains an Apache implementation of java-based NLP tools.

- This package can perform a variety of NLP functions including:

  - Sentence Splitting
  - Tokenization
  - Part of Speech Tagging (POS)
  - Named Entity Recognition
  - Chunking
  - Parsing

```
#############################################################
# Initialize the packages.
#############################################################

# Make sure that we are running 32 bit version of R!!!!

library("openNLP")
library("openNLPdata")
library("tm")
library("NLP")
library("openNLPmodels.en")

#####################################################################
# Create some text to manipulate with NLP
#####################################################################

s <- paste(c("Pierre Vinken, 61 years old, will join the board as
             "nonexecutive director on November 29. ",
             "Mr. Vinken is chairman of Chicago , ",
             "the Dutch publishing group."),
           collapse = "")

# s <- as.String(mydata$Message)

s <- as.String(s)
```

Natural Language Processing in R

Sentence splitting

- Sentence boundary = period + space(s) + capital letter

- **Example:** Unusually, the gender of crocodiles is determined by temperature. If the eggs are incubated tat over 33c, then the egg hatches into a male or 'bull' crocodile. At lower temperatures only female or 'cow' crocodiles develop.

- Unusually, the gender of crocodiles is determined by temperature.
- If the eggs are incubated tat over 33c, then the egg hatches into a male or 'bull' crocodile.
- At lower temperatures only female or 'cow' crocodiles develop.

```
##############################################################
# Sentence Splitting
##############################################################

# Break apart the text into separate sentances.

sent_token_annotator <- Maxent_Sent_Token_Annotator()
a1 <- annotate(s, sent_token_annotator)
```

Sentence	Text
1	Pierre Vinken, 61 years old, will join the board as a nonexecutiv
2	Mr. Vinken is chairman of Chicago , the Dutch publishing group.

Natural Language Processing

Tokenization

- Convert a sentence into a sequence of tokens
- Divides the text into smallest units (usually words), removing punctuation.

- **Example:** A Saudi Arabian woman can get a divorce if her husband doesn't give her coffee.

- A Saudi Arabian woman can get a divorce if her husband does n't give her coffee .

```
##########################################################
# Tokenization
##########################################################

# Find the individual words in each sentence.

word_token_annotator <- Maxent_Word_Token_Annotator()
word_token_annotator
a2 <- annotate(s, word_token_annotator, a1)
a2
```

ID	Type	Start	End
3	word	1	6
4	word	8	13
5	word	14	14
6	word	16	17
7	word	19	23
8	word	25	27
9	word	28	28

Natural Language Processing in R

Part-of-speech tagging

- Assign a part-of-speech tag to each token in a sentence.
- Example: Most lipstick is partially made of fish scales

- Most/ JJS lipstick/ NN is/ VBZ partially/ RB made/ VBN of/ IN fish/ NN scales/ NNS

Tokens with POS Tag		
Pierre/NNP	Vinken/NNP	,/,
61/CD	years/NNS	old/JJ
,/,	will/MD	join/VB
the/DT	board/NN	as/IN
a/DT	nonexecutive/JJ	director/NN
on/IN	November/NNP	29/CD
./.	Mr./NNP	Vinken/NNP
is/VBZ	chairman/NN	of/IN
Chicago/NNP	,/,	the/DT
Dutch/JJ	publishing/NN	group/NN
./.		

```
##################################################
# Part of Speech Tagging
##################################################

pos_tag_annotator <- Maxent_POS_Tag_Annotator()
pos_tag_annotator
a3 <- annotate(s, pos_tag_annotator, a2)
a3

## Variant with POS tag probabilities as (additional) features.
head(annotate(s, Maxent_POS_Tag_Annotator(probs = TRUE), a2))

## Determine the distribution of POS tags for word tokens.
a3w <- subset(a3, type == "word")
tags <- sapply(a3w$features, `[[`, "POS")
tags
table(tags)

## Extract token/POS pairs (all of them): easy.
sprintf("%s/%s", s[a3w], tags)
```

Natural Language Processing

- Part of Speech Tags

Tag	Description	Tag	Description
CC	Coordinating conjunction	PRP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential *there*	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	to
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

Natural Language Processing

Named Entity Recognition

- Named entity recognition classify tokens in text into predefined categories such as date, location, person, time.
- The name finder can find up to seven different types of entities - date, location, money, organization, percentage, person, and time.
- **Example:** Diana Hayden was in Philadelphia on October 3rd.

- <namefind/person> Diana Hayden </namefind/person> was in<namefind/location> Philadelphia </namefind/location> on <namefind/date> October 3rd </namefind/date>

```
##############################################################
# Named Entity Recognition
##############################################################

# requires package openNLPmodels.en
# from http://datacube.wu.ac.at

## Entity recognition for persons.
entity_annotator <- Maxent_Entity_Annotator(kind="person")
entity_annotator
annotate(s, entity_annotator, a2)
```

ID	Type	Start	End	Features
34	entity	1	13	kind = person
34	entity	119	125	kind = location
34	entity	80	90	kind = date

Hierarchy	Text
Name	Pierre Vinken
Location	Chicago
Date	November 29

Natural Language Processing

Chunking (shallow parsing)

- the identification of parts of speech and short phrases (like noun phrases). Part of speech tagging tells you whether words are nouns, verbs, adjectives, etc., but it doesn't give you any clue about the structure of the sentence or phrases in the sentence.

- In NER, your goal is to find named entities, which tend to be noun phrases (though aren't always), so you would want to know that President Barack Obama is in the following sentence:

- President Barack Obama criticized insurance companies and banks as he urged supporters to pressure Congress to back his moves to revamp the health-care system and overhaul financial regulations.

- But you wouldn't necessarily care that he is the subject of the sentence.

```
##############################################################
# Chunker - Shallow Parsing
##############################################################
## Chunking needs word token annotations with POS tags.

sent_token_annotator <- Maxent_Sent_Token_Annotator()
word_token_annotator <- Maxent_Word_Token_Annotator()
pos_tag_annotator <- Maxent_POS_Tag_Annotator()
a3 <- annotate(s,
               list(sent_token_annotator,
                    word_token_annotator,
                    pos_tag_annotator))

annotate(s, Maxent_Chunk_Annotator(), a3)
annotate(s, Maxent_Chunk_Annotator(probs = TRUE), a3)
```

ID	Type	Start	End	Features
3	word	1	6	POS=NNP, chunk_tag=B-NP, chunk_prob=0.9740431
4	word	8	13	POS=NNP, chunk_tag=I-NP,chunk_prob=0.9816025
5	word	14	14	POS=,, chunk_tag=O, chunk_prob=0.9863059
6	word	16	17	POS=CD, chunk_tag=B-NP,chunk_prob=0.9926662
7	word	19	23	POS=NNS, chunk_tag=I-NP,chunk_prob=0.9854421
8	word	25	27	POS=JJ, chunk_tag=B-ADJP,chunk_prob=0.9978292
9	word	28	28	POS=., chunk_tag=O, chunk_prob=0.9909762
10	word	30	33	POS=MD, chunk_tag=B-VP, chunk_prob=0.979816
11	word	35	38	POS=VB, chunk_tag=I-VP,chunk_prob=0.9857121
12	word	40	42	POS=DT, chunk_tag=B-NP,chunk_prob=0.9932718

Example: He reckons the current account deficit will narrow to only 1.8 billion in September

NP VP NP VP PP NP PP NP

Natural Language Processing

Tree Bank Parsers

- A program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb.
- Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences.
- Their development was one of the biggest breakthroughs in natural language processing in the 1990s.

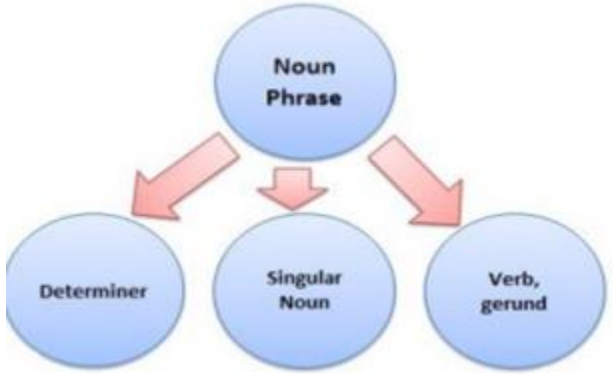
```
########################################################################
# Parse Annotator
########################################################################

## Need sentence and word token annotations.
sent_token_annotator <- Maxent_Sent_Token_Annotator()
word_token_annotator <- Maxent_Word_Token_Annotator()
a2 <- annotate(s, list(sent_token_annotator, word_token_annotator))

parse_annotator <- Parse_Annotator()
## Compute the parse annotations only.
p <- parse_annotator(s, a2)
## Extract the formatted parse trees.
ptexts <- sapply(p$features, `[[`, "parse")
ptexts
```

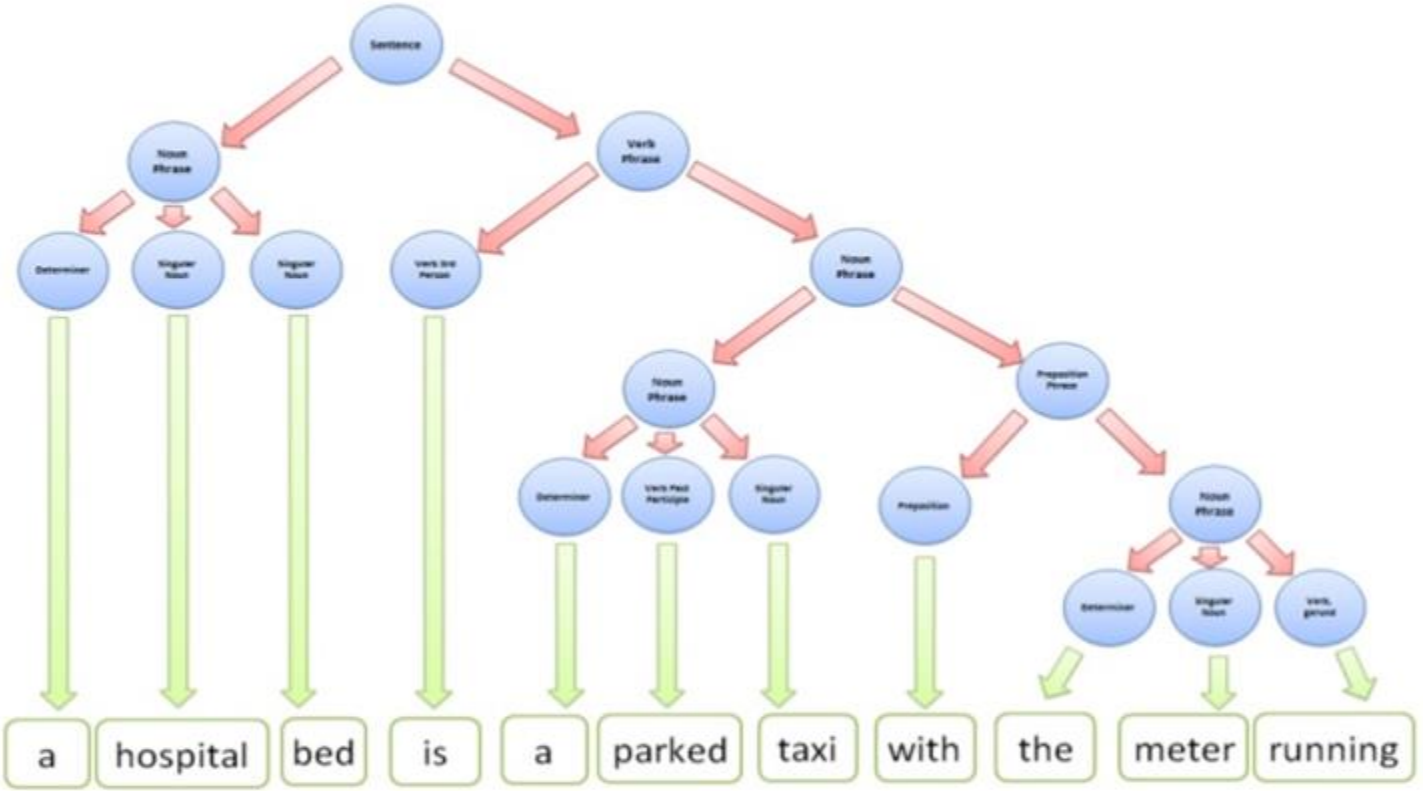
```
[1] "(TOP (S (NP (NP (NNP Pierre) (NNP Vinken))(, ,) (ADJP (NP (CD 61) (
NNS years)) (JJ old)))(, ,) (VP (MD will) (VP (VB join) (NP (DT the) (NN
board)) (PP (IN as) (NP (NP (DT a) (JJ nonexecutive) (NN director)) (PP
(IN on) (NP (NNP November) (CD 29)))))))(. .)))"
[2] "(TOP (S (NP (NNP Mr.) (NNP Vinken)) (VP (VBZ is) (NP (NP (NP (NN ch
airman)) (PP (IN of) (NP (NNP Chicago)))) (, ,) (NP (DT the) (JJ Dutch)
(NN publishing) (NN group))))(. .)))"
```

- Example: A hospital bed is a parked taxi with the meter running.

- (TOP (S (NP (DT A) (NN hospital) (NN bed)) (VP (VBZ is) (NP (NP (DT a) (VBN parked) (NN taxi)) (PP (IN with) (NP (DT the) (NN meter) (VBG running)))))))

Natural Language Processing



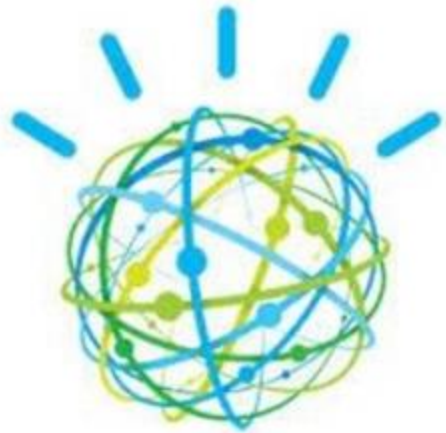



```
## Read into NLP Tree objects.
ptrees <- lapply(ptexts, Tree_parse)
ptrees
```

```
(TOP
  (S
    (NP (NNP Mr.) (NNP Vinken))
    (VP
      (VBZ is)
      (NP
        (NP (NP (NN chairman)) (PP (IN of) (NP (NNP Chicago))))
        (, ,)
        (NP (DT the) (JJ Dutch) (NN publishing) (NN group))))
    (. .)))
```

Natural Language Processing

- It's ironic that natural language, the symbol system that is easiest for humans to learn and use, is hardest for a computer to master.
- Long after machines have proven capable of inverting large matrices with speed and grace, they still fail to master the basics of our spoken and written languages.
- Eventually you will be able to address your computer as though you were addressing another person.
- This goal is not easy to reach. "Understanding" language means, among other things, knowing what concepts a word or phrase stands for and knowing how to link those concepts together in a meaningful way.
- NLP remains one the uncharted frontiers in computer science and constantly evolving.



Text Analytics Example: Clustering Uncategorized RSS Feeds

News Sites on the WWW

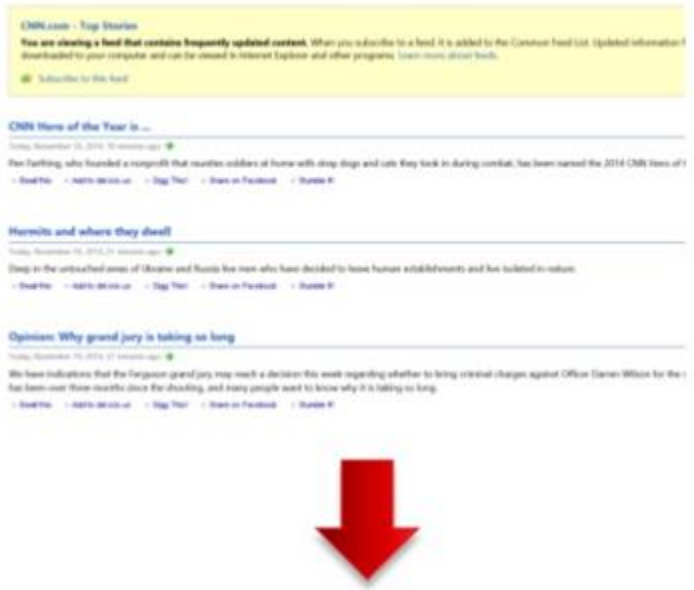
- The content of major news outlets can be a rich source of information for text analytical work.
- Data obtained from the news can be leveraged to help predict stock movements as well as a drive new regression coefficients that can be used in data mining / predictive analytics.
- RSS feeds are an XML format designed to make the extraction of text information from the news easy for BI systems.
- This presentation will process various RSS news feeds obtained by BBC, CNN, NBC News, and NY Times and we will cluster them for categorization using an unsupervised learning technique.

Acquiring the Data

- The data has been scrapped into MS SQL through an SSIS package designed to find major RSS news feeds.

- News Page

- RSS Feed



Title	Description	News Feed	Extraction Date
Surprise! Catcher is soldier dad	Soldier Ian Jones returns from a deployment in Afghanistan	CNN	4/27/2014
Why this 23-year-old has 24 kids	It's a sunny April afternoon at the University of Rwanda Coll	CNN	4/27/2014
Check for fifties under the mattress	There are still millions of Houblon £50 notes in circulation, d	BBC	4/28/2014
It looks like a puppy, but it's not ...	A family was surprised to learn the abandoned 'puppy' they	CNN	4/28/2014
New Dove beauty ad goes too far?	Dove's latest viral ad campaign has hurt a lot of feelings. Ke	CNN	4/28/2014
Outrage: 'What's the captain doing?'	A teen's father gave a South Korean TV network footage fro	CNN	4/28/2014
A swift punishment, but is it a just one	The debate over the banning of LA Clippers' Donald Sterling	BBC	4/29/2014
Amanda Knox Rejects Court's Reason	Amanda Knox rejected an Italian court's contention Tuesday	NBCNews	4/29/2014
Cast Is Announced for Next 'Star War	The seventh installment of the space epic will feature seven	NY Times	4/29/2014
Heroin and Alcohol Led to the Deaths	Autopsies of Jeffrey Reynolds and Mark Kennedy, the forme	NY Times	4/29/2014
Moscow Journal: Amid a Revived Eas	At the Museum of the Cold War, visitors are drawn as much	NY Times	4/29/2014

Data Preparation for Prediction

- We will split the dataset into a training and testing sample. We will sample 70% for our training data and 30% for the validation test.

- Training Dataset

city	court	crimea	murder	officials	police	president	south	state
1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	0	2	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0

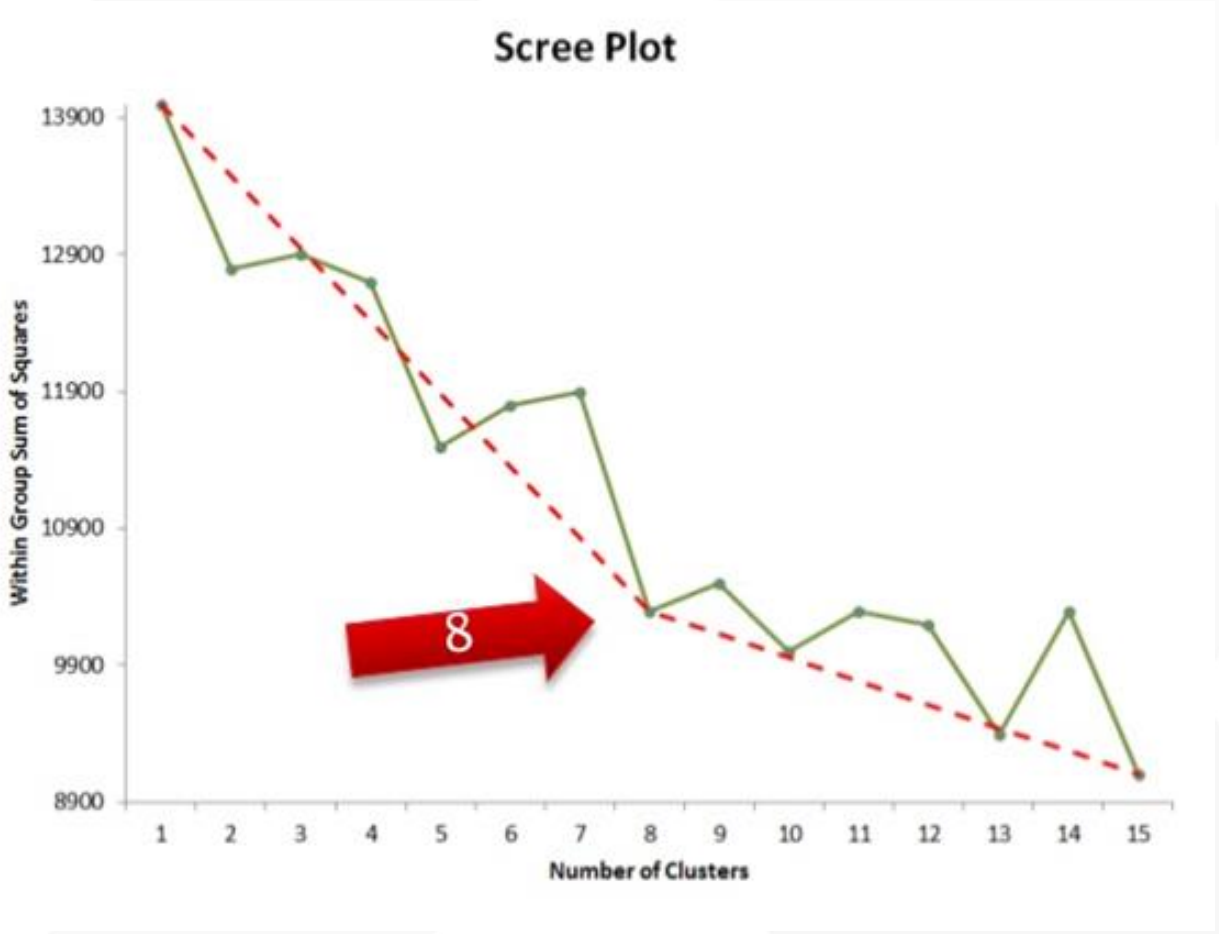
- Testing Dataset

city	court	crimea	murder	officials	police	president	south	state
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0

- We will implement a kNN algorithm to perform the clustering of the training documents.

Creating the unsupervised clustering

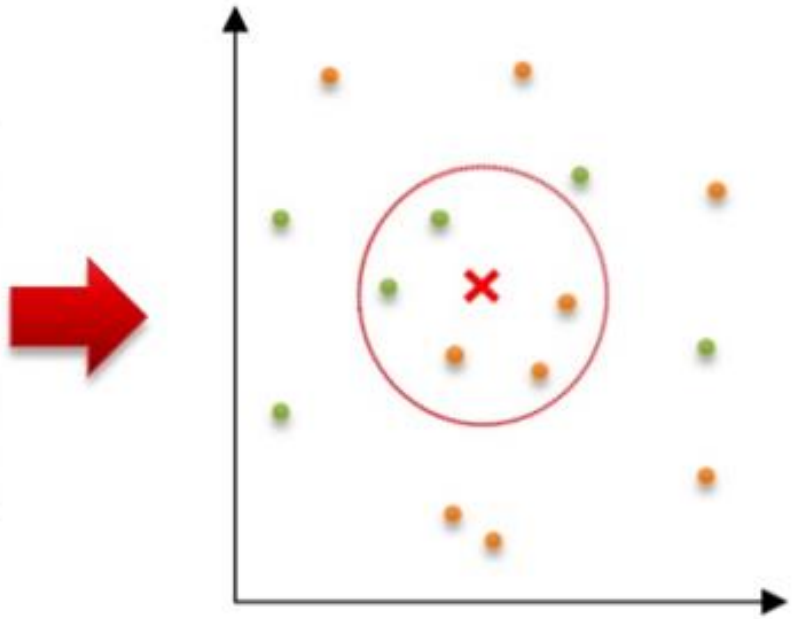
- In order to tune the classification parameter for the kNN, we will produce and review a scree plot.
- A bend at the elbow indicates the optimal parameters for the clustering procedure.
- The scree plot indicates that k = 8 might be an appropriate starting parameter.



Creating the unsupervised cluster

- The testing dataset was passed through a kNN classification algorithm.

city	court	crimea	murder	officials	police	president	south	state
1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	0	2	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0



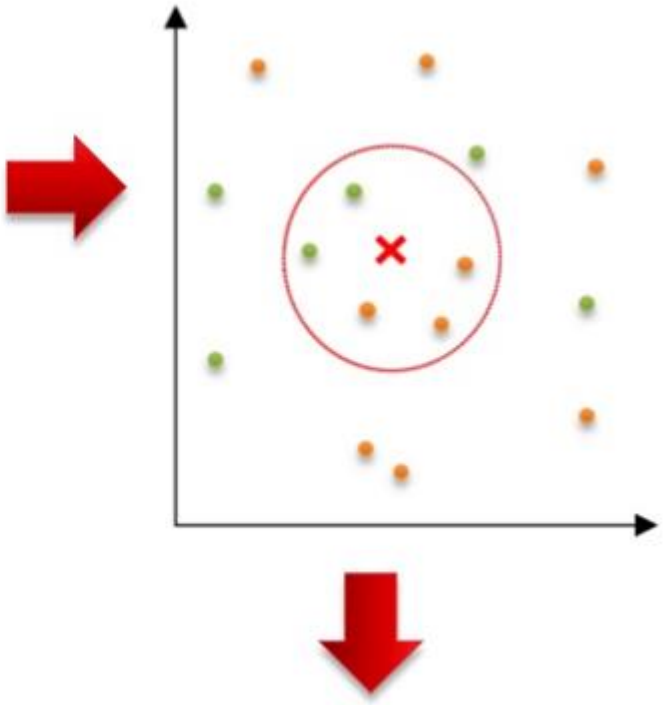
Cluster Review

- Once the 8 clusters have been produced we can review the underlying terms to detect any overarching theme(s).
- Understanding these terms will allow for us to create meaningful labels to the clusters.
- Drawing from other text data preparation techniques, such as NLP, it is possible to created unsupervised computer generated cluster labels.

Cluster	Description	Document Count	% of Total
1	Opinions	1532	12.3%
2	Global Conflict	1252	10.1%
3	General	4989	40.2%
4	US Regional	642	5.2%
5	Crime	451	3.6%
6	Coming Events	1249	10.1%
7	Police	930	7.5%
8	Political	1361	11.0%

Applying the cluster on New RSS Feeds

city	court	crimea	murder	officials	police	president	south	state
1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	0	2	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0



Cluster	city	court	crimea	murder	officials	police	president	south	state
General	1	0	0	0	0	0	0	0	0
General	0	0	0	0	0	0	0	0	1
US Regional	0	0	0	0	0	0	0	1	1
Political	0	0	0	0	0	0	1	0	0
Coming Events	0	0	0	0	0	0	0	0	0
General	0	0	0	0	0	0	0	0	0
Global Conflict	1	0	2	0	1	0	0	0	0
Crime	0	0	0	1	0	0	0	0	0
Police	0	1	0	0	0	1	0	0	0
Opinions	0	0	0	0	0	0	0	0	0

Text Analytics Example: Social Media Sentiment & Network Analysis

Social Media Analysis

- The purpose of this tutorial is to showcase how we can build off of the text analytics techniques we have discussed so far and tap into a wealth of data made available from social media platforms.
- We will focus on pulling in data from Facebook and Twitter through the R interface.
- Additionally, we will perform a couple of additional analysis including:
  - Social Network Diagrams
  - Natural Language Processing
  - Sentiment Analysis
  - Word Cloud Visualization
- Discussion of business applications for these techniques.

Acquiring the data from Facebook

- R makes it possible to easily pull information related to Facebook and Twitter through their API. This makes R extremely appealing as the primary statistical tool to integrate unstructured and structured data analysis.


```
#############################################################
# Facebook Data Extraction
#############################################################

library(Rfacebook)
library(Rook)

fb_oauth <- fbOAuth(app_id="XXXXXXXXXXXXXXX",
                    app_secret="YYYYYYYYYYYYYY")
load("fb_oauth")

# Pull Friend Demogrpahics, Likes, Check-In,& News Feed

my_friends <- getFriends(token=fb_oauth, simplify=TRUE)
Likes <-getLikes(user="1075852970", n=500, token=fb_oauth)
Checkin <-getCheckins(user="1075852970", n=10,
                      token=fb_oauth, tags = FALSE)
NewsFeed <- getNewsfeed(token=fb_oauth, n=500)
```

Name	Wall Post	Gender	Birthday	Relationship Status
Kimberly Barnett	Happy Halloween! Great time with family and Friends	Female	NA	NA
Russ Kelsey	Gonna be a long night preparing for russells 1st bday party...hope to see everyon	Male	5/25/14	Married
James Martin	#stupidrandomfacti can do 22 handstand pushups before i collapse upon myself	Male	12/31/82	Engaged
Eric Alford	Hawks score and my buzzer remote takes a crap.	Male	9/17/80	Married
Stephen Lejeune	Looking forward to seeing my art buddys and some sweet ass bands at my favori	Male	NA	Married
Rob Kolb	Thanks to all for the birthday wishes! The love of my life, Marcey Bender Moorn	Male	10/14/65	In a relationship
JoAnne Serowka	OMG! They are reporting SNOW in McHenry & Woodstock- Ugh- here come the	Female	3/3/55	Married
Michael Schindler	The most impressive building I have ever seen	Male	7/14/14	NA
Tommy Brodie	Happy Halloween, especially to CPD who wouldn't let me surf.	Male	9/8/14	NA
Kelly Wulf Kellerman	Happy Birthday!!! Hope you have a great day today!! Good luck on your run!!	Female	9/16/81	Married
Eric Morgenstern	Gotta love it, my 3 year old son gets in the car in the morning and wants to listen	Male	10/8/81	Married
Susan L. Tarson	Passed recertification so I am once again a National Board Certified Teacher! Th	Female	8/5/14	In a relationship
Marcy Tunison	I'm blaming this blustery snowy Halloween on all the little Elsa's out there today	Female	5/28/85	Married
Greg Franczyk	So Dealer day at the show was interesting. I ran into a guy who collects Disney W	Male	NA	Single
April Stoltman	It kind of saddened me to not see kids out today.....we only passed 2 groups of k	Female	4/18/14	Married
April Stoltman	It's my favorite day of the year! Have a safe fabulous candy filled holiday!	Female	4/18/14	Married
Jennifer Murphy	Update on the car - the insurance adjusted told us today that they are going to g	Female	5/3/72	Married

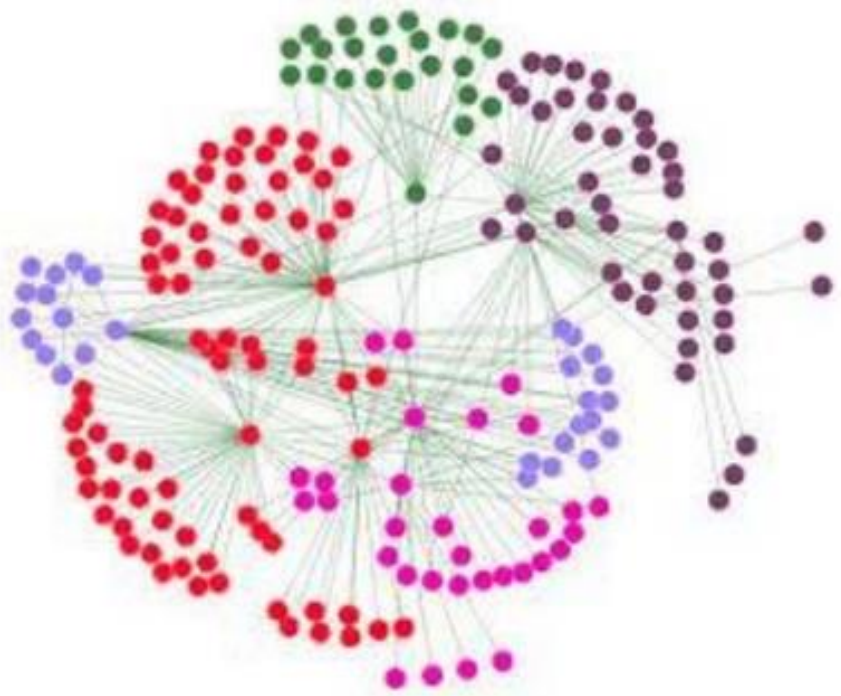
Acquiring the data from Twitter

```
#################################################################
# Twitter Data Extraction
#################################################################

library("twitteR")

requestURL <- "https://api.twitter.com/oauth/request_token"
accessURL <- "https://api.twitter.com/oauth/access_token"
authURL <- "https://api.twitter.com/oauth/authorize"
consumerKey <- "XXXXXXXXXXXXXXXXX"
consumerSecret <- "YYYYYYYYYYYYYYYYYYYYYYYYY"
Cred <- OAuthFactory$new(consumerKey=consumerKey,
                         consumerSecret=consumerSecret,
                         requestURL=requestURL,
                         accessURL=accessURL,
                         authURL=authURL)

# Pull Tweets with #Adrian Peterson

Tweets <- searchTwitter('#AdrianPeterson',
                        n=100, cainfo="cacert.pem")
```

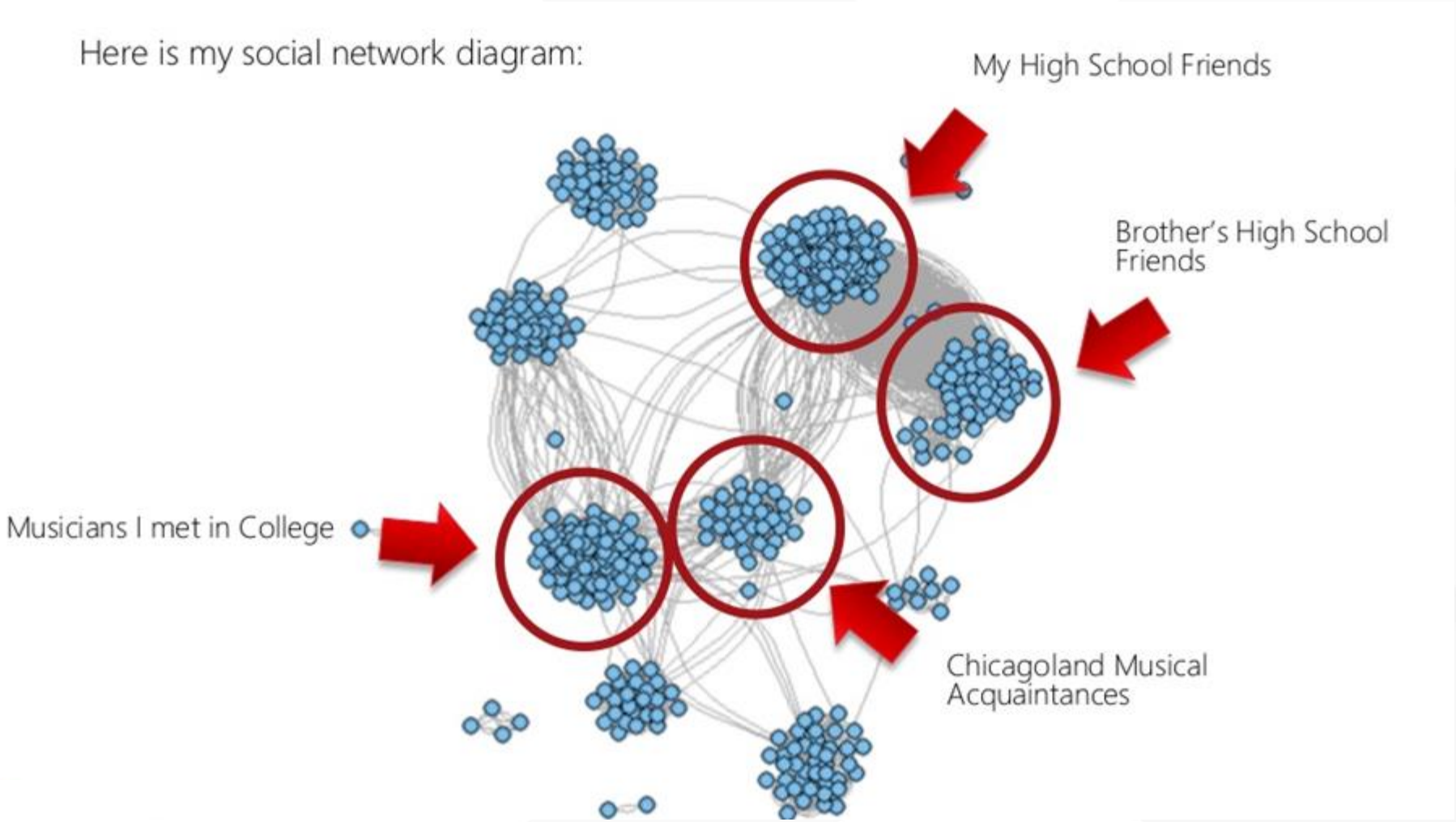
Date	Name	Tweet	retweetCount
11/18/14	DariusHoward_PR	#AdrianPeterson is suspended for the remainder of the season w/o pay.	0
11/18/14	Young_Hypocrite	RT @TheRoot: #AdrianPeterson is suspended from the @NFL without pa	11
11/18/14	stroker66ace	@FCC please look into the actions of @1057FMTheFan. I believe they ar	0
11/18/14	Adorable_Mikey	Watched my buddy hit his wife and get fired. Hit my kid and got the same	0
11/18/14	LilBill2345	Whether you agree with physical child discipline or not you have to admit	0
11/18/14	DJRobertHorry	homeboy is a dum-dum, but the NFLPA is going to sue the fuck out of the	0
11/18/14	bwolfe23	RT @AceKlubKasanova: My thoughts #NFL #AdrianPeterson http://t.co/S	1
11/18/14	legalspeaks	RT @BringMN: All the developments in the #AdrianPeterson suspension,	2
11/18/14	ItsShanaRenee	NEW! New Rules: #RogerGoodell doesn't fight fair in #AdrianPeterson's s	0
11/18/14	KrismaerS3	#AdrianPeterson Not sure having an angry unemployed without pay man	0
11/18/14	Dj_Ango_	Ha "@fishsports: Do we grasp the irony in #AdrianPeterson thinking he's	0
11/18/14	ReddFoxxThePoet	Hm rt"@thetoyman1: #AdrianPeterson has been suspended without pay	0
11/18/14	The_Rob_Wagner	The only people dumber than #AdrianPeterson are the morons defending	0

Social Network Analysis - Facebook

- Now that we have some data to analyze. Lets start off by creating a Social Network Analysis and diagram.
- Social network analysis (SNA) is the use of network theory to analyze social networks.
- Social network analysis views social relationships in terms of network theory, consisting of nodes, representing individual actors within the network, and ties which represent relationships between the individuals, such as friendship, kinship, organizations and sexual relationships.



Natural Language Processing



What is Sentiment Analysis?

- Sentiment analysis is software for automatically extracting opinions, emotions, and sentiments in text.
- It allows for us to track attitudes and feelings on the web. People write blog posts, comments, reviews, and tweets about all sorts of different topics.
- We can track products, brands, and people for example and determine if they are viewed positively or negatively on the web.

Sentiment Analysis

- It allows for businesses to track:
  - Flame Detection (bad rants)
  - New Product Perception
  - Brand Perception
  - Reputation Management
- It allows individuals to get
  - An opinion on something (reviews) on a global scale.

Why use Sentiment Analysis?

- According to a presentation by NetBase CMO Lisa Joy Rosner, the average consumer mentions specific brands over 90 times per week in conversations with friends, family, and co-workers.
- In addition, 53% of people on Twitter recommend companies and/or products in their tweets, with 48% of them delivering on their intention to buy the product.
- This means that Twitter and other social media are a perfect complement to traditional market research – especially has usage has spread through more demographics (social networking use among internet users aged 50+ has nearly doubled to 42% last year).
- You get unbiased, more truthful thoughts and opinions, and the target consumers come to you, naturally, and for free.

Sentiment Analysis – Multiple Areas

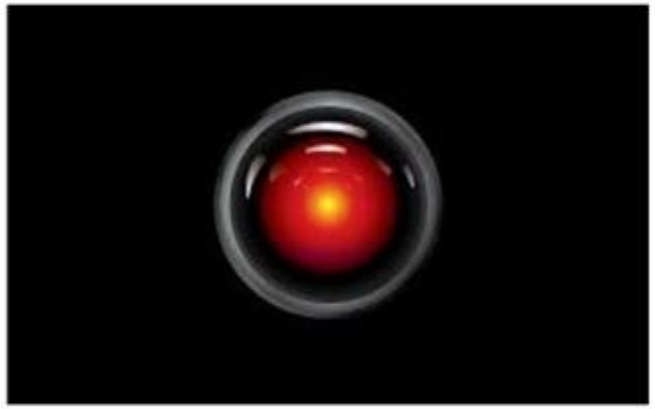
Natural Language Processing

- NLP deals with the actual text element. It transforms it into a format that the machine can use.

Artificial Intelligence

- It uses information given by the NLP and uses a lot of maths to determine whether something is negative or positive; it is used for clustering.

Sentiment Analysis



The problem has several dimensions:

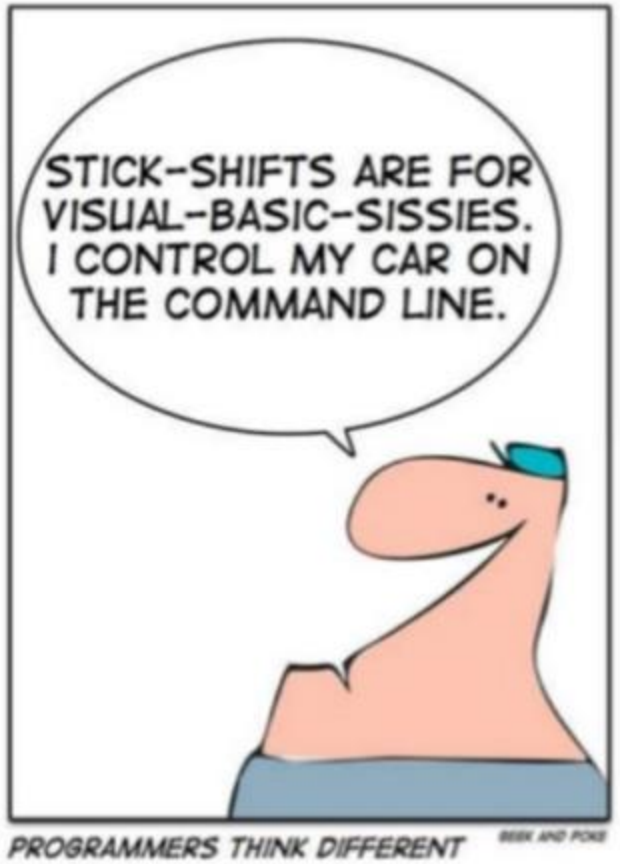
- How does a machine define subjectivity & sentiment?
- How does a machine analyze polarity (negative/positive)?
- How does a machine deal with subjective word senses?
- How does a machine assign an opinion rating?
- How does a machine know about sentiment intensity?

Sentiment Analysis

- It is not always easy to differentiate between fact and opinion.
- An opinion to a machine is called the "quintuple" (Bing Liu).

$$(o_j, f_{jk}, so_{ijkl}, h_i, t_i)$$

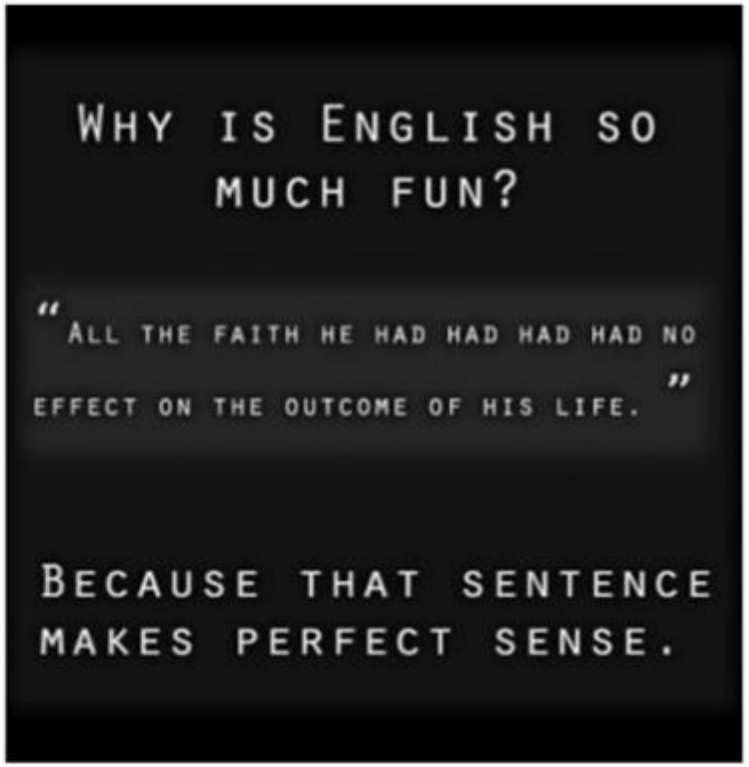
- o_j = The thing in question (ex. Product)
- f_{jk} = a feature of o_j
- so_{ijkl} = the sentiment value of the opinion of the opinion holder hi on feature f_{jk} of object o_j at time t_i
- These 5 elements have to be identified by the machine.
- All of these problems are unresolved by computer science and are open areas ripe for advancement.



Sentiment Analysis

Language is ambiguous. Consider the following:

- "The watch isn't water resistant" – In a product review this could be negative.
- "As much use as a trap door on a lifeboat" – negative but not obvious to the machine.
- "The canon camera is better than the Fisher Price on" – comparisons are hard to classify.
- "imo the ice cream is luuuurrrrrrrrvely" – slang and the way we communicate in general needs to be processed.

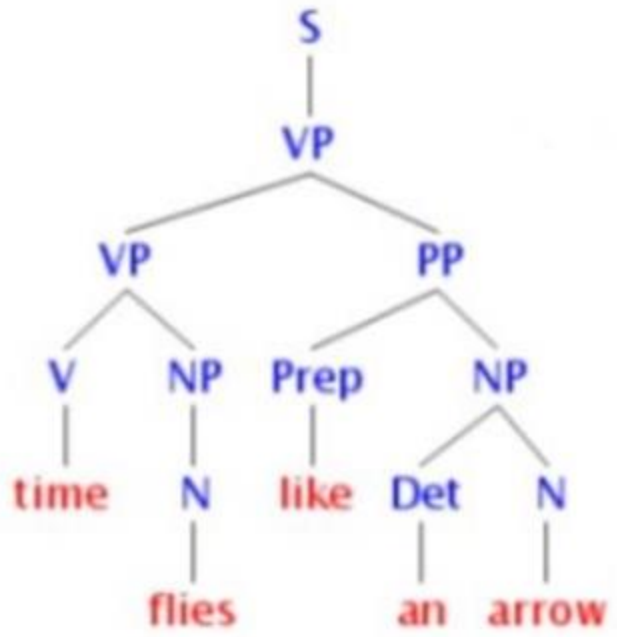


Sentiment Analysis – Natural Language Processing

Part of Speech Tagging

- The word in the text (or the sentence) at tagged used a POS tagger so that it assigns a label to each word, allowing the machine to do something like this:

S = subject
VP = Verb Phrase
V = Verb
N = Noun
NP = Noun Phrase
PP = Preposition
Det = Determiner



- Then we extract defined patterns like [Det] + [N] for example

Sentiment Analysis – Natural Language Processing

- We also look at the sentiment orientation (SO) of the patterns we extracted. For example, we may have extracted:

- Amazing + phone

which is:

- [JJ] + [NN] (or adjective followed by a noun in human)

- The opposite might be "Terrible" for example. In this stage, the computer tries to situate the words on an emotive scale.

Sentiment Analysis Scoring

The average sentiment orientation of all the phrases we gathered is computed.

- For a particular entry (Facebook Post / Tweet) the entry will be positive if the total count of positive terms is greater than the count of negative terms and vice versa.
- The sentiment scores can then be aggregated to calculate the score of the entire corpus:

Corpus Score = Positive Instances / Total Instances

- This allows the machine to say something like:

- "Generally people like the new iphone" ---> They recommend it.
- "Generally people hate the new iphone" ---> They do not recommend it.

Does Sentiment Analysis Work?

- The wider you throw your net and the more complex the language, the less accurate the system will be. This is simply due to the complexity the machine has to deal with.
- If you want to classify sentiments into + / - groups, then you are more likely to get a good result than if you are trying to classify into more exact groups (Excellent, incredible, good, etc...).
- More granularity requires more accuracy and this in turn requires a deeper understanding of the human language.

Facebook Sentiment Analysis

How is the sentiment determined for these Facebook posts?

Name	Wall Post	Gender	Birthday	Relationship Status
Kimberly Barnett	Happy Halloween! Great time with family and Friends	Female	NA	NA
Russ Kelsey	Gonna be a long night preparing for russells 1st bday party...hope to see everyon	Male	5/25/14	Married
James Martin	#stupidrandomfacti can do 22 handstand pushups before i collapse upon myself	Male	12/31/82	Engaged
Eric Alford	Hawks score and my buzzer remote takes a crap.	Male	9/17/80	Married
Stephen Lejeune	Looking forward to seeing my art buddys and some sweet ass bands at my favori	Male	NA	Married
Rob Kolb	Thanks to all for the birthday wishes! The love of my life, Marcey Bender Moorm	Male	10/14/65	In a relationship
JoAnne Serowka	OMG! They are reporting SNOW in McHenry & Woodstock- Ugh- here come the	Female	3/3/55	Married
Michael Schindler	The most impressive building I have ever seen	Male	7/14/14	NA
Tommy Brodie	Happy Halloween, especially to CPD who wouldn't let me surf.	Male	9/8/14	NA
Kelly Wulf Kellerman	Happy Birthday!!! Hope you have a great day today!! Good luck on your run!!	Female	9/16/81	Married
Eric Morgenstern	Gotta love it, my 3 year old son gets in the car in the morning and wants to listen	Male	10/8/81	Married
Susan L. Tarson	Passed recertification so I am once again a National Board Certified Teacher! Th	Female	8/5/14	In a relationship
Marcy Tunison	I'm blaming this blustery snowy Halloween on all the little Elsa's out there today	Female	5/28/85	Married
Greg Franczyk	So Dealer day at the show was interesting. I ran into a guy who collects Disney W	Male	NA	Single
April Stoltman	It kind of saddened me to not see kids out today.....we only passed 2 groups of k	Female	4/18/14	Married
April Stoltman	It's my favorite day of the year! Have a safe fabulous candy filled holiday!	Female	4/18/14	Married
Jennifer Murphy	Update on the car - the insurance adjusted told us today that they are going to g	Female	5/3/72	Married

Example

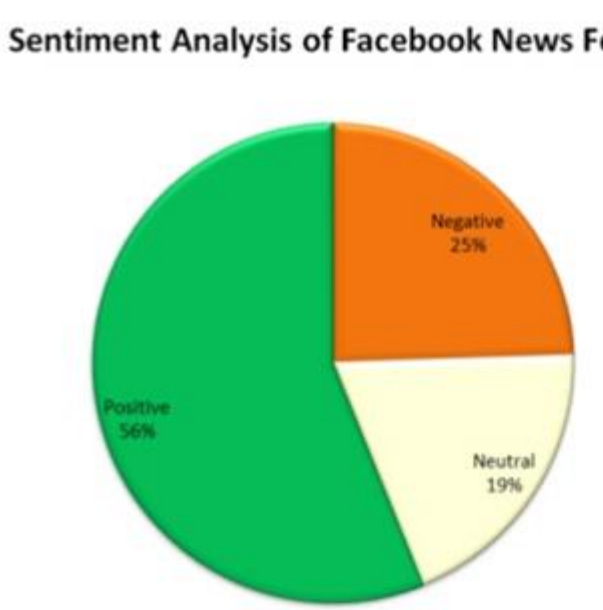
- Wall Post: "Happy Halloween! Great time with family and friends"
  - Natural Language Processing Rules: Happy = [Adj] and Great time = [Adj] + [V]
  - Positive Sentiment: 2, Negative Sentiment = 0
  - Sentiment Score = Positive

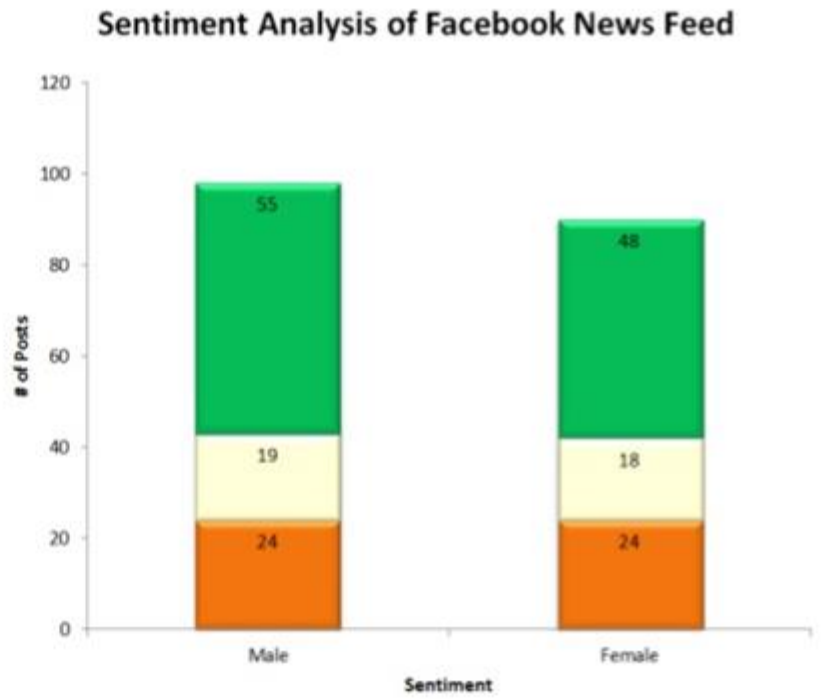
Facebook Sentiment Analysis

Wall Post	Sentiment
Happy Halloween! Great time with family and Friends	Positive
Gonna be a long night preparing for russells 1st bday party...hope to see everyon	Negative
#stupidrandomfacti can do 22 handstand pushups before i collapse upon myself	Positive
Hawks score and my buzzer remote takes a crap.	Positive
Looking forward to seeing my art buddys and some sweet ass bands at my favori	Positive
Thanks to all for the birthday wishes! The love of my life, Marcey Bender Moorm	Positive
OMG! They are reporting SNOW in McHenry & Woodstock- Ugh- here come the	Negative
The most impressive building I have ever seen	Positive
Happy Halloween, especially to CPD who wouldn't let me surf.	Positive
Happy Birthday!!! Hope you have a great day today!! Good luck on your run!!	Positive
Gotta love it, my 3 year old son gets in the car in the morning and wants to listen	Positive
Passed recertification so I am once again a National Board Certified Teacher! Th	Neutral
I'm blaming this blustery snowy Halloween on all the little Elsa's out there today	Negative
So Dealer day at the show was interesting. I ran into a guy who collects Disney W	Neutral
It kind of saddened me to not see kids out today.....we only passed 2 groups of ki	Negative
It's my favorite day of the year! Have a safe fabulous candy filled holiday!	Positive
Update on the car - the insurance adjusted told us today that they are going to g	Neutral

Facebook Sentiment Analysis

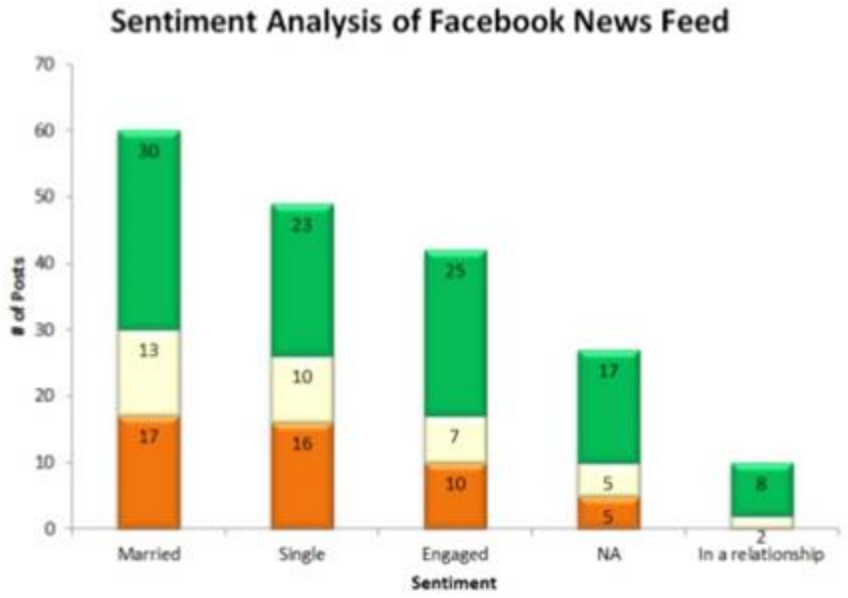
- Here is a breakdown of the sentiment scoring for my Facebook newsfeed that contains 188 posts that occurred on 10/31/2014.

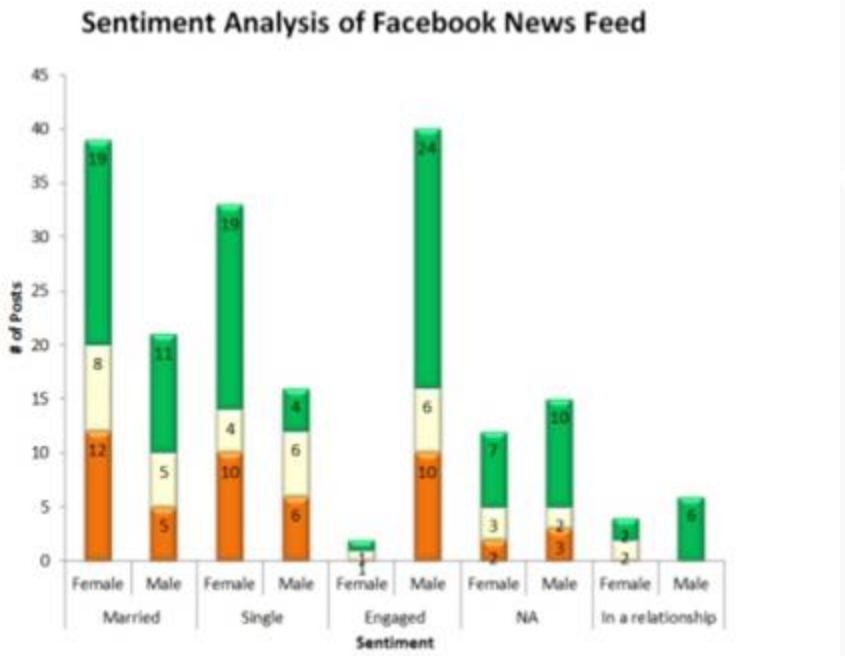




- We can see that there is 98 updates from males and 90 from females. In total, the algorithm had scored 56% posts as positive versus 25% negative.

Facebook Sentiment Analysis





- Married & Engaged friends have the highest degree of positive sentiment within their postings.
- Females who are married or single seem to be the most positive. There are almost no females who are engaged in this cohort.
- Interestingly, engaged males are slightly more positive than married males based off of their posts.

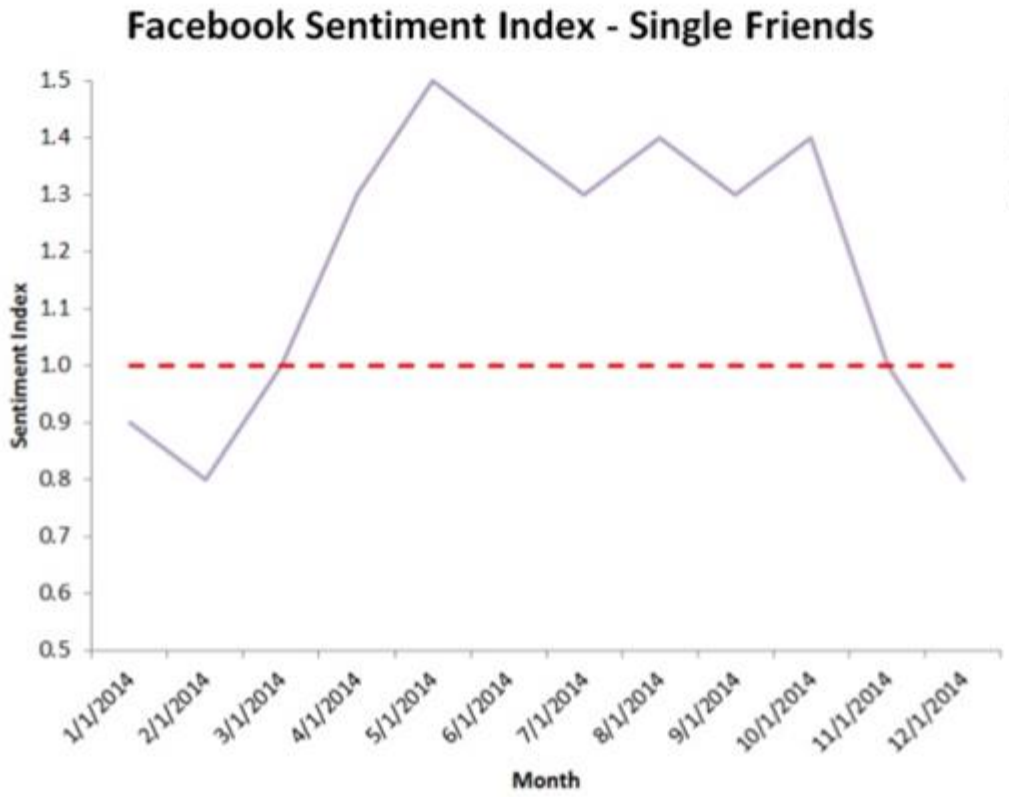
Facebook Sentiment Analysis

- How can we leverage these sentiment scores to incorporate the knowledge gained into predictive model building?
- Sentiment Index = Positive / Negative

Marital Status	Negative	Positive	Sentiment Index
Married	17	30	1.8
Single	16	23	1.4
Engaged	10	25	2.5
NA	5	17	3.4
In a relationship	1	8	8.0

Date	Marital Status	Sentiment Index
1/1/2014	Single	0.9
2/1/2014	Single	0.8
3/1/2014	Single	1.0
4/1/2014	Single	1.3
5/1/2014	Single	1.5
6/1/2014	Single	1.4
7/1/2014	Single	1.3
8/1/2014	Single	1.4
9/1/2014	Single	1.3
10/1/2014	Single	1.4
11/1/2014	Single	1.0
12/1/2014	Single	0.8
etc...	etc...	etc...

Facebook Sentiment Analysis



Business Application:
Product mentions instead of single friends

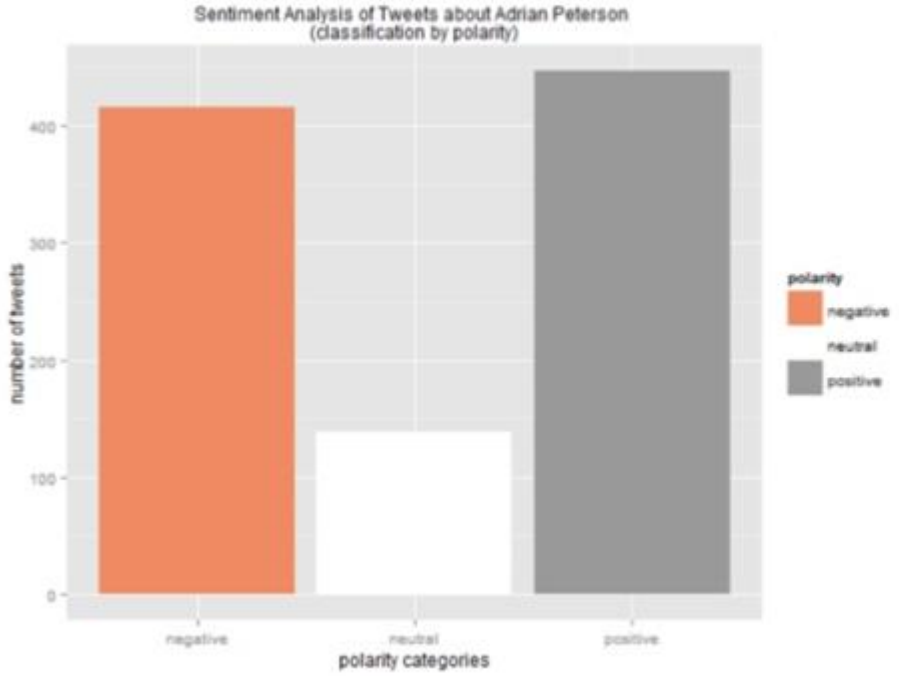
- This example graph shows that there is a negative sentiment during the colder periods of time and around the major family holidays (Thanksgiving, Christmas, St. Valentines Day)
- Once the weather warms up in Chicago, the overall sentiment of these singles improves.

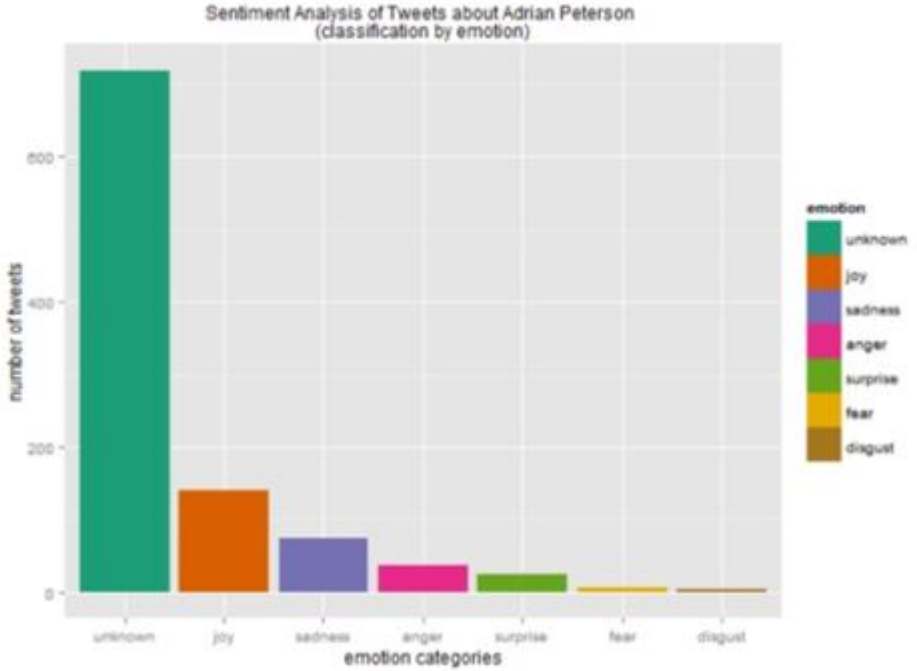
Facebook Sentiment Analysis

- A word cloud is valuable tool used to present a visual image of the magnitude of the individual terms used throughout the overall postings.
- We will present two separate word clouds for review: total terms and sentiment based.

Twitter Sentiment Analysis

- An analysis of twitter posts reveal that a slim majority of posts have a positive sentiment for the #AdrianPeterson hash tag.





Twitter Sentiment Analysis

Business Applications of Sentiment Analysis

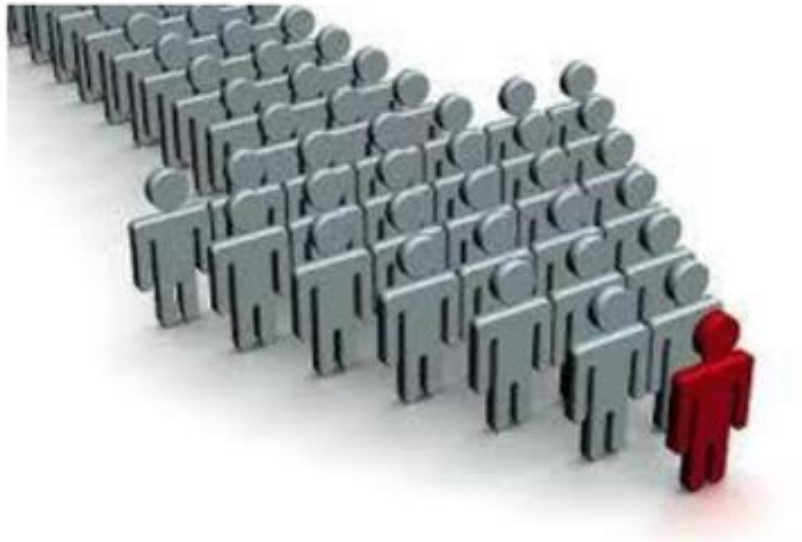
- One of the most important areas for sentiment analysis, and social media monitoring in general, is bridging the gap between insight and action.
- It's one thing to retrieve a sentiment pie chart. It's another to masterfully place it within the context of your brand's social media performance.

Business Applications of Sentiment Analysis

- The key to successful engagement is sentiment prioritization:

- **Influence:** Because social media mentions are plentiful, prioritization tools must continue evolving. Of the 10,000 tweets and blog posts about your brand, how do you pick the top 50 to focus on?

- Example: If you need to neutralize the mentions that hurt your brand the most, you should drill down into negative mentions, identify the content coming from the most influential people in your industry, understand how far each tweet traveled, and how many people were impacted by this content.



Business Applications of Sentiment Analysis

- **Reputation:** Each notable user should have a social media reputation profile. If someone's negative sentiment indexes higher than average (i.e. that person hates everything equally), then that person's negative sentiment should be somewhat discounted.

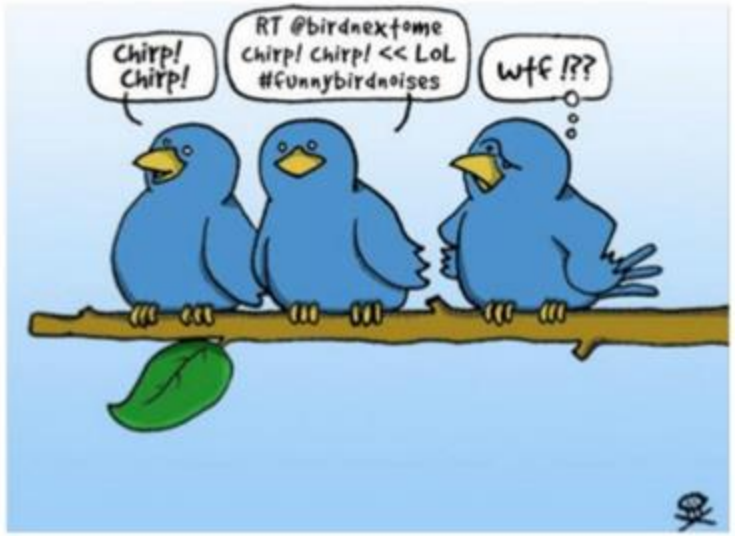
- **Intensity:** As far as sentiment algorithms are concerned, part of a successful prioritization process is going to be identifying the intensity of each mention. "I really hate product X and will never buy it" is quite different from "Product X is running a little slow today." Ability to cross-reference intensity, influence, trajectory, velocity and sentiment of each social media mention will drive us towards a reliable priority system.

- **Isolating content types:** A lot of social media mentions are neutral in nature and some social media sources tend to skew higher on the neutral scale. For example, a higher percentage of updates are neutral on Twitter than any other medium (consider these common examples: "I just had a cup of coffee," or "craving tacos for lunch – who's in?" or "Apple launches the iPad tablet"). Depending on the source you are looking at, your sentiment results will differ — this should be expected. Make sure your sentiment platform allows you to isolate results by content type.

Building off of Sentiment Analysis

- **Sentiment override:** Because automated sentiment is not going to be 100% accurate, you, the user, need to have some kind of override control. When picking a tool, ensure that it allows you to override sentiment, and toss irrelevant results.

- **Entity level vs. article level sentiment:** Until recently, the industry default has been able to measure sentiment at the level of the article. Over time, some platforms have developed ways to measure sentiment on the level of the entity (entity level analysis can measure the sentiment of an entity or multiple entities within an article even if the overall sentiment of the article is different).



Text Analytics Example: The Blog's Writer Gender

Gender Analysis from Blog Entries

- One appealing aspect of text analytics is the ability to create a profile of characteristics for individuals related to our written and spoken text.
- We have a compiled list of list of blog entries where the gender of the person is known and we would like predict the gender of blogs where the gender is unknown.
- Understanding characteristics/ demographics of individuals has a wide variety of applications in terrorism detection, counterintelligence, education, business, and fraud.

Gender Analysis Blog Entries

- Each of the blogs will be processed converting the unstructured text data into a structured table with each row relating to a specific blog.

Text	Gender
I'm back from vacation, and still digging my way out of everything that's piled up while I've been offline. While I catch up, I thought I'd share with you a demo that Eric Iverson was gracious enough to share with me. It uses Yahoo! BOSS to support an exploratory search experience on top of a general web search engine. When you perform a query, the application retrieves a set of related term candidates using Yahoo's key terms API. It then scores each term by dividing its occurrence count within the result set by its global occurrence count–a relevance measure similar to one my former colleagues and I used at Endeca in enterprise contexts. You can try out the demo yourself at http://www.ittybittysearch.com/. While it has rough edges, it produces nice results–especially considering the simplicity of the approach. Here's an example of how I used the application to explore and learn something new. I started with ["information retrieval"]. I noticed "interactive information retrieval" as a top term, so I used it to refine. Most of the refinement suggestions looked familiar to me–but an unfamiliar name caught my attention: "Anton Leuski". Following my curiosity, I refined again. Looking at the results, I immediately saw that Leuski had done work on evaluating document clustering for interactive information retrieval. Further exploration made it clear this is someone whose work I should get to know–check out his home page! I can't promise that you'll have as productive an experience as I did, but I encourage you to try Eric's demo. It's simple examples like these that remind me of the value of pursuing HCIR for the open web.	Male
Who moved my Cheese??? The world has been developing in and out in all the areas and to create a difference in this competitive world... we need to change... change the way we take our things... but we rather change or atleast try to change..... we try the same routine work evryday and expect to get more and when things fail such as losing a job, loss in business we would upset, discouraged, frustrated and keep on hanging to the same thing again and start complaining. CHANGE IS GOOD.... LETS WELCOME IT..!!! wondering wat is all about Cheese?? and what actually is all about " Who Moved My Cheese??? " Well...!!!! Who moved my cheese?? is a simple parable that reveals profound thoughts. It is an enlightening story of four characters who live in a maze and look for cheese to nourish them and make them happy. The story is about two mice called "SNIFF" and "SCURRY" and two little men ... smaller in size and who were similar to us people. Their names were "HEM" and "HAW" Cheese is a metaphor for what you want to have in life - whether it is a good job, loving relationship, money or a possession, health or spiritual peace of mind.And the maze is where you look for what you want - the organisation you work in, or the family or community you live in. Everyday both the mice and men spent time in the maze looking for their own special cheese. The mice had only Rodent brains while the men used their brains, filled with many beliefs. The common thing between the rodents and these men is tat every morning they went in search for the cheese.	Female

Gender Analysis from Blog Entries

- We need to first construct a corpus (a collection of texts) using the dataset.
- Then we will apply a looping function for the data preparation and apply the cleaning function to blog entry.

```
#################################################################################
# Corpus Cleanup
#################################################################################

Corpus.mydata <- tm_map(Corpus.mydata, removeNumbers)
Corpus.mydata <- tm_map(Corpus.mydata, tolower)
Corpus.mydata <- tm_map(Corpus.mydata, removeWords, stopwords("english"))
Corpus.mydata <- tm_map(Corpus.mydata, removePunctuation)
Corpus.mydata <- tm_map(Corpus.mydata, stripWhitespace)

# Matrix with columns as the terms and rows as the documents.
Corpus.TDM <- DocumentTermMatrix(Corpus.mydata)

# Remove Sparse Terms from Corpus.TDM
Corpus.TDM <- removeSparseTerms(Corpus.TDM, 0.95)
```

Create a Term Document Matrix

Blogger	Gender	already	also	although	always	amazing	another	anyone	anything
1	Male	0	0	0	1	0	0	0	0
2	Male	0	0	0	0	0	0	0	0
3	Male	1	2	0	0	0	0	0	0
4	Male	0	1	0	0	0	0	0	0
5	Female	0	0	0	2	0	0	0	0
6	Male	0	0	1	1	1	0	0	0
7	Female	0	0	0	0	0	2	0	0
8	Male	2	0	0	0	0	0	0	3
9	Female	0	0	0	1	0	0	0	0
10	Female	1	0	0	0	0	0	0	0
11	Male	1	2	0	0	0	1	0	0
12	Male	1	3	0	0	0	2	0	1
13	Female	2	1	0	0	1	0	0	1
14	Male	0	3	0	0	0	0	0	0
15	Male	0	5	0	1	0	0	1	1
16	Female	0	0	0	0	0	1	0	0
17	Female	0	0	0	0	0	0	0	0
18	Male	0	0	1	0	1	0	0	0
19	Male	0	1	0	0	0	0	0	1
20	Female	0	0	0	2	0	0	0	0
21	Male	0	1	0	0	0	0	0	0
22	Female	0	1	0	0	0	0	0	0
23	Male	0	0	0	0	0	0	0	2
24	Male	0	0	0	0	0	0	0	0
25	Female	0	0	0	0	0	0	0	0

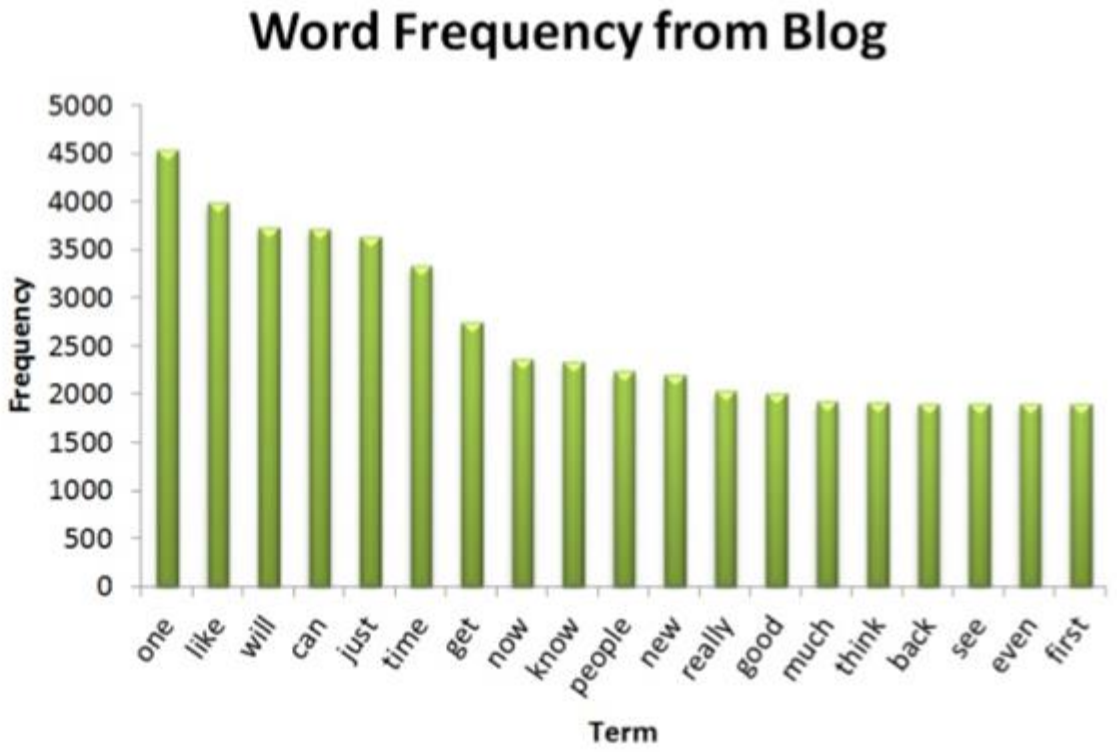
- The Gender Field contains the known gender of each blogger.
- There are 430 terms to the right and a frequency count of each terms occurrence.
- This list of terms was pared down using a sparsity threshold parameter of 0.95.

Associations & Frequency Terms

- We can gain some insight and intelligence by understanding frequencies and associations of terms.

- **Example:** Which words are associated with the term "company" ?

Search Term: Company	
Related	**Association**
order	0.76
person	0.58
case	0.45
business	0.41
may	0.41
without	0.39
right	0.31
without	0.39
right	0.31



Data Preparation for Prediction

- We will split the dataset into a training and testing sample. We will sample 70% for our training data and 30% for the validation test.

- Training Dataset

Gender	already	also	although	always	amazing	another	anyone	anything
Male	0	0	0	1	0	0	0	0
Male	0	0	0	0	0	0	0	0
Male	1	2	0	0	0	0	0	0
Male	0	1	0	0	0	0	0	0
Female	0	0	0	2	0	0	0	0
Male	0	0	1	1	1	0	0	0
Female	0	0	0	0	0	2	0	0
Male	2	0	0	0	0	0	0	3
Female	0	0	0	1	0	0	0	0
Female	1	0	0	0	0	0	0	0
Male	1	2	0	0	0	1	0	0
Male	1	3	0	0	0	2	0	1
Female	2	1	0	0	1	0	0	1
Male	0	3	0	0	0	0	0	0
Male	0	5	0	1	0	0	1	1

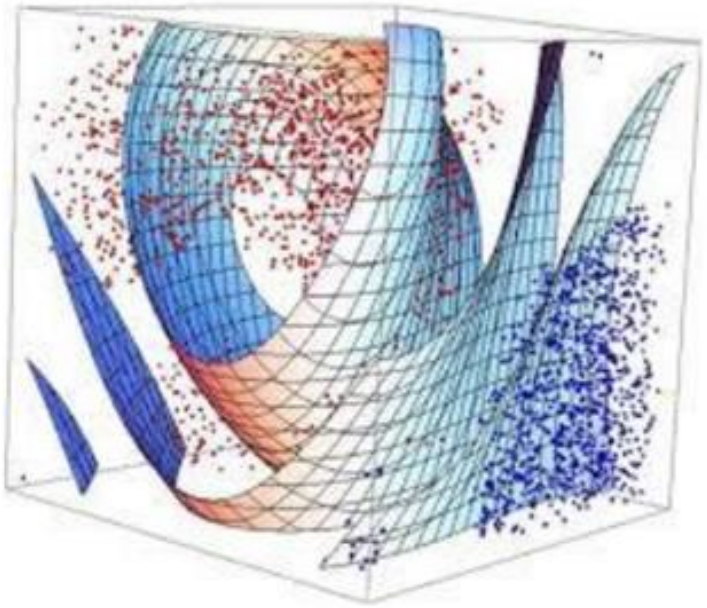
- Testing Dataset

Blogger	already	also	although	always	amazing	another	anyone	anything
1	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0
3	0	0	1	0	1	0	0	0
4	0	1	0	0	0	0	0	1
5	0	0	0	2	0	0	0	0
6	0	1	0	0	0	0	0	0
7	0	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	2
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0

- The Gender column has been removed from the testing set because we will be predicting the gender based off of the blog entries.

Build Support Vector Machine

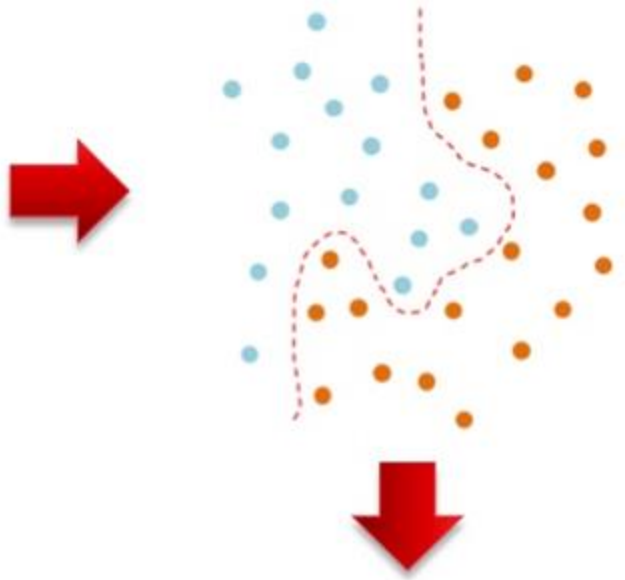
- First, lets run a tuning function to identify the best parameters to use for the SVM model.
- The process runs a 10-fold cross validation methodology and identified the following:
  - Gamma = 0.001
  - Cost = 10
- Best Performance = 0.3879642



Testing the support vector Machine

- We created a random test sample of the dataset and included only the measurement variables. This data was not involved in the initial training of the Support Vector Machine but will be used to validate the results from passing the data through the SVM.

Gender	already	also	although	always	amazing	another	anyone	anything
Male	0	0	0	1	0	0	0	0
Male	0	0	0	0	0	0	0	0
Male	1	2	0	0	0	0	0	0
Male	0	1	0	0	0	0	0	0
Female	0	0	0	2	0	0	0	0
Male	0	0	1	1	1	0	0	0
Female	0	0	0	0	0	2	0	0
Male	2	0	0	0	0	0	0	3
Female	0	0	0	1	0	0	0	0
Female	1	0	0	0	0	0	0	0
Male	1	2	0	0	0	1	0	0
Male	1	3	0	0	0	2	0	1
Female	2	1	0	0	1	0	0	1
Male	0	3	0	0	0	0	0	0
Male	0	5	0	1	0	0	1	1



Prediction Accuracy:
53.2%

Blogger	Gender	Prediction
1	Male	Male
2	Male	Female
3	Female	Female
4	Male	Male
5	Female	Female
6	Male	Male
7	Male	Female
8	Male	Female
9	Female	Male
10	Male	Female

Lessons Learned

- 53.2% is not a strong prediction based off of the data. It is only 3.2% better than randomly guessing the gender.
- It is important to understand that unstructured text analysis does not always yield strong predictive results.
- This is due in part of having:
  - Insufficient sample size
  - Lack of pattern in the blog entries.
  - Tuning stemming technique
  - Data dictionary definition
  - Incorrect Predictive Algorithm / parameters
- Nevertheless, we should be able to improve this result through a careful review of the modeling techniques and altering our approach when necessary. As our text analytics improve over time, so will our capabilities for developing robust profiles of individuals.

Text Analytics World – New Trends

Text and Data: Two Way Street

- New types of applications
  - New ways to make sense of data, enrich data
- Harvard – Analyzing Text as Data
  - Detecting deception, Frame Analysis
- Narrative Science – take data (baseball statistics, financial data) and turn into a story
- Political campaigns using Big Data, social media, and text analytics
- Watson for healthcare – help doctors keep up with massive information overload

Integration of Text and Data Analytics

- Expertise Location: Case Study: Data and Text
- Data Sources:
  - HR Information: Geography, Title-Grade, years of experience, education, projects worked on, hours logged, etc.
- Text Sources:
  - Document authored (major and minor authors) – data and/or text
  - Documents associated (teams, themes) – categorized to a taxonomy
  - Experience description – extract concepts, entities
- Self-reported expertise – requires normalization, quality control
- Complex judgments:
  - Faceted application
  - Ensemble methods – combine evaluations

Social Media: Beyond Simple Sentiment

- Beyond Good and Evil (positive and negative)
  - Social Media is approaching next stage (growing up)
  - Where is the value? How get better results?
- Importance of Context – around positive and negative words
  - Rhetorical reversals – “I was expecting to love it”
  - Issues of sarcasm, (“Really Great Product”), slanguage
- Limited value of Positive and Negative
  - Degrees of intensity, complexity of emotions and documents
  - Addition of focus on behaviors – why someone calls a support center – and likely outcomes
  - Trending analysis – sentiment just one dimension

Social Media: Beyond Simple Sentiment

- Two basic approaches [Limited accuracy, depth]
  - Statistical Signature of Bag of Words
  - Dictionary of positive & negative words
- Essential – need full categorization and concept extraction to get full value from social media
- New Taxonomies – Appraisal Groups – Adjective and modifiers – “not very good”
  - Four types – Attitude, Orientation, Graduation, Polarity
  - Supports more subtle distinctions than positive or negative
- Emotion taxonomies - Joy, Sadness, Fear, Anger, Surprise, Disgust
  - New Complex – pride, shame, embarrassment, love, awe
  - New situational/transient – confusion, concentration, skepticism

Information Platform: Beyond Search

- Why Text Analytics?
  - Enterprise search has failed to live up to its potential
  - Enterprise Content management has failed to live up to its potential
  - Taxonomy has failed to live up to its potential
  - Adding metadata, especially keywords has not worked
  - BI, CI limited sources //labor intensive// SBA need language
- What is missing?
  - Intelligence – human level categorization, conceptualization
  - Semantics, language –not technology
- Text Analytics can be the foundation that (finally) drives success – search, content management, and much more

Information Platform: Tagging Documents

- How do you bridge the gap – taxonomy to documents?
- Tagging documents with taxonomy nodes is tough
  - And expensive – central or distributed
- Library staff –experts in categorization not subject matter
  - Too limited, narrow bottleneck
  - Often don't understand business processes and business uses
- Authors – Experts in the subject matter, terrible at categorization
  - Intra and Inter inconsistency, "intertwingleness"
  - Choosing tags from taxonomy – complex task
  - Folksonomy – almost as complex, wildly inconsistent
  - Resistance – not their job, cognitively difficult = non-compliance

Information Platform: Content Management

- Hybrid Model
  - Publish Document -> Text Analytics analysis -> suggestions for categorization, entities, metadata - > present to author
  - Cognitive task is simple -> react to a suggestion instead of select from head or a complex taxonomy
  - Feedback – if author overrides -> suggestion for new category
  - Facets – Requires a lot of Metadata - Entity Extraction feeds facets
- Hybrid – Automatic is really a spectrum – depends on context
  - All require human effort – issue of where and how effective
- External Information - human effort is prior to tagging
  - Build on expertise – librarians on categorization, SME's on subject terms

Building on the Platform Expertise Analysis

- Expertise Characterization for individuals, communities, documents, and sets of documents
- Experts prefer lower, subordinate levels
  - Novice & General – high and basic level
- Experts language structure is different
  - Focus on procedures over content
- Applications:
  - Business & Customer intelligence – add expertise to sentiment
  - Deeper research into communities, customers
  - Expertise location- Generate automatic expertise characterization based on documents

Behavior Prediction – Telecom Customer Service

- Problem – distinguish customers likely to cancel from mere threats
- Basic Rule
  - (START_20, (AND, (DIST_7,"[cancel]", "[cancel-what-cust]"),
  - (NOT,(DIST_10, "[cancel]", (OR, "[one-line]", "[restore]", “[if]”)))))
- Examples:
  - customer called to say he will **cancell** his **account** **if** the does not stop receiving a call from the ad agency.
  - cci and **is upset that he has the asl charge** and **wants it off** **or** her is going to **cancel** his act
- More sophisticated analysis of text and context in text
- Combine text analytics with Predictive Analytics and traditional behavior monitoring for new applications

Building on the Platform Variety of New Applications

- Essay Evaluation Software - Apply to expertise characterization
  - Avoid gaming the system – multi-syllabic nonsense
    - Model levels of chunking, procedure words over content
- Legal Review
  - Significant trend – computer-assisted review (manual =too many)
  - TA- categorize and filter to smaller, more relevant set
  - Payoff is big – One firm with 1.6 M docs – saved $2M
- Financial Services
  - Trend – using text analytics with predictive analytics – risk and fraud
  - Combine unstructured text (why) and structured transaction data (what)
  - Customer Relationship Management, Fraud Detection
  - Stock Market Prediction – Twitter, impact articles

Building on the Platform for Pronoun Analysis: Fraud Detection; Enron Emails

- Patterns of “Function” words reveal wide range of insights
- Function words = pronouns, articles, prepositions, conjunctions, etc.
  - Used at a high rate, short and hard to detect, very social, processed in the brain differently than content words
- Areas: sex, age, power-status, personality – individuals and groups
- Lying / Fraud detection: Documents with lies have
  - Fewer and shorter words, fewer conjunctions, more positive emotion words
  - More use of “if, any, those, he, she, they, you”, less “I”
  - More social and causal words, more discrepancy words
- Current research – 76% accuracy in some contexts
- Text Analytics can improve accuracy and utilize new sources
- Data analytics (standard AML) can improve accuracy

Future Directions Need for Vision and Quick Application

- Text Analytics is weird, a bit academic, and not very practical
  - It involves language and thinking and really messy stuff
- On the other hand, it is really difficult to do right (Rocket Science)
- Every application is a custom job – no standards, generality
  - Trend: Modules and Templates (Logic & Data separate)
- Organizations don't know what text analytics is and what it is for
- TAW Survey shows - need two things:
  - Strategic vision of text analytics in the enterprise
  - Real life functioning program showing value and demonstrating an understanding of what it is and does
- Quick Start – Strategic Vision – Software Evaluation – POC / Pilot

Adding Intelligence – High Level

- Understand your customers
  - What they are talking about and how they feel about it
- Empower your employees
  - Not only more time, but they work smarter
- Understand your competitors
  - What they are working on, talking about
  - Combine unstructured content and rich data sources – more intelligent analysis
- Integration of all of the above – Platform
  - Integration at the semantic level

Building the Platform - Strategic Vision

- Info Problems – what, how severe
- Formal Process - KA audit – content, users, technology, business and information behaviors, applications - Or informal for smaller organization,
- Contextual interviews, content analysis, surveys, focus groups, ethnographic studies, Text Mining
- Category modeling – Cognitive Science – how people think
  - Monkey, Panda, Banana
- Natural level categories mapped to communities, activities
    - Novice prefer higher levels
    - Balance of informative and distinctiveness
- Text Analytics Strategy/Model – What is text analytics?

New Directions in Text Analytics: Conclusions

- Text Analytics is poised for explosive growth
- Text Analytics is a platform / infrastructure
- New models are opening up
  - Beyond sentiment – emotion & behavior, cognitive science
  - Enterprise Hybrid Model, Data and Text models
- Big obstacles remain
  - Strategic Vision of text analytics in the enterprise, applications
  - Concrete and quick application to drive acceptance
  - Software still too complex, un-integrated
- New types of applications opening up new frontiers
  - Text Intelligence and artificial intelligence – the promise?
- This is a great time to get into text analytics!

Contact Us

Visit us on: http://www.analytixlabs.in/

For more information, please contact us: http://www.analytixlabs.co.in/contact-us/

Or email: info@analytixlabs.co.in

Call us we would love to speak with you: (+91) 99105-09849

Join us on:

Twitter - http://twitter.com/#!/AnalytixLabs

Facebook - http://www.facebook.com/analytixlabs

LinkedIn - http://www.linkedin.com/in/analytixlabs

Blog - http://www.analytixlabs.co.in/category/blog/