UNIVERSITÀ DEGLI STUDI DI BARI
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Informatica

Accesso intelligente all'informazione

Prof. Giovanni Semeraro
Dott. Pasquale Lops
Dott. Marco Degemmis
***<lastname>*@di.uniba.it**

Corso di Gestione della Conoscenza d'Impresa
Anno Accademico 2005-2006

Outline

- Document/Text Mining: From Text to Knowledge
  - ✓ Definizione
  - ✓ Data mining vs. Text mining
  - ✓ Perchè Text mining?
- Intelligent Information Retrieval
  - ✓ Boolean and Vector Space Retrieval Models
  - ✓ Integrazione di conoscenza lessicale: WordNet
  - ✓ Metriche per la valutazione
- Text Categorization
  - ✓ Machine Learning for Text Categorization
  - ✓ Metriche per la valutazione
- Esempi di sistemi ed applicazioni: apprendimento di profili utente nelle biblioteche elettroniche

UNIVERSITÀ DEGLI STUDI DI BARI
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Informatica

Document/Text Mining: From Text to Knowledge

Prof. Giovanni Semeraro
Dott. Pasquale Lops
Dott. Marco Degemmis

Corso di Gestione della Conoscenza d'Impresa
Anno Accademico 2005-2006

Gestire la conoscenza

... significa:

- Raccogliere la conoscenza
- Organizzarla (strutturarla, classificarla)
- Distribuirla
- Renderla accessibile a chi ne ha bisogno (nel momento e nel posto in cui serve)

...al fine di:

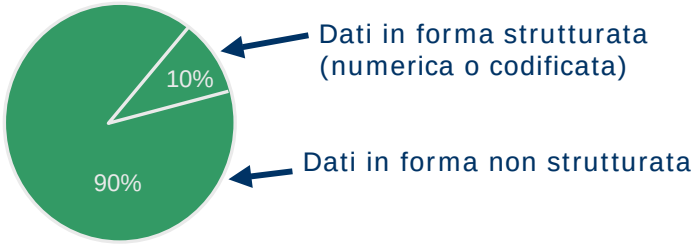
- Risparmiare tempo
- Migliorare la qualità dei servizi
- Ridurre i tempi di accesso all'informazione ed alla fruizione dei servizi

Dati-Informazione-Conoscenza

La conoscenza è un capitale:

- intangibile
- volatile
- difficile da concretizzare e conservare

Circa il 90% dei dati presenti nei database del mondo è in forma non strutturata



Automatic Knowledge Management

- Obiettivi
  - ✓ Costruzione di sistemi in grado di processare documenti in linguaggio naturale
  - ✓ Acquisizione / Ritrovamento di conoscenza da basi di dati in forma testuale

Acquisizione della Conoscenza

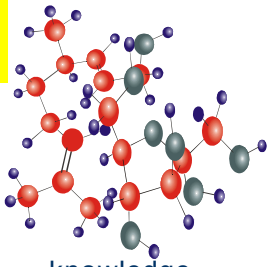
"From Text to Knowledge"

ONTOLOGY
CONSTRUCTION

Text

- Document classification
- Information Extraction
- Text mining

Document-based
ontology definition



knowledge

Outline

- Knowledge Discovery from Text: Text Mining
  - ✓ Definizione
  - ✓ Data mining vs. Text mining
  - ✓ Perchè Text mining?

"Search" versus "Discover"

	Search (goal-oriented)	Discover (opportunistic)
Structured Data	Data Retrieval	Data Mining
Unstructured Data (Text)	Information Retrieval	Text Mining

Data Retrieval

- Ritrovamento di record in un database strutturato.

Database Type	Structured
Search Mode	Goal-driven
Atomic entity	Data Record
Example Information Need	"Find a Japanese restaurant in Boston that serves vegetarian food."
Example Query	"SELECT * FROM restaurants WHERE city = boston AND type = japanese AND has_veg = true"

Information Retrieval

- Cerca informazione rilevante in una sorgente di dati non strutturati (tipicamente in formato testo)

Database Type	
Search Mode	
Atomic entity	
Example Informa Need	ıese restaurant in Boston ːs vegetarian food."
Example Query	"Japanese restaurant Boston" or Boston->Restaurants->Japanese

Data Mining

- Scopre nuova conoscenza attraverso l'analisi di dati

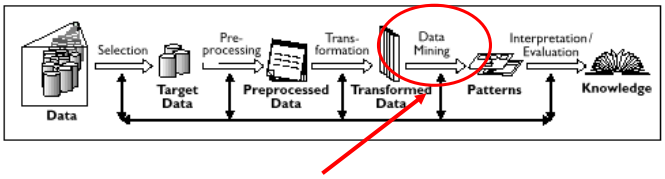
Database Type	Structured
Search Mode	Opportunistic
Atomic entity	Numbers
Example Information Need	"Show trend over time in # of visits to Japanese restaurants in Boston "

The KDD Process

Knowledge Discovery from Databases

"The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"

Usama Fayyad, Gregory Piatetsky-Shapiro and Padhraic Smyth, 1996. Knowledge Discovery and Data Mining: Towards a Unifying Framework. In Proceedings of The Second Int. Conference on Knowledge Discovery and Data Mining, pages 82—88.



Note: data mining is just one step in the process

Data Mining@work

Factual

CustomerId	LastName	FirstName	BirthDate	Gender
0721134	Doe	John	11/17/1945	Male
0721168	Brown	Jane	05/20/1963	Female
0730021	Adams	Robert	06/02/1959	Male

Transactional

CustomerId	Date	Time	Store	Product	CouponUsed
0721134	07/09/1993	10:18am	GrandUnion	WheatBread	No
0721134	07/09/1993	10:18am	GrandUnion	AppleJuice	Yes
0721168	07/10/1993	10:29am	Edwards	SourCream	No
0721134	07/10/1993	07:02pm	RiteAid	LemonJuice	No
0730021	07/10/1993	08:34pm	Edwards	SkimMilk	No
0730021	07/10/1993	08:34pm	Edwards	AppleJuice	No
0721168	07/12/1993	01:13pm	GrandUnion	BabyDiapers	Yes
0730021	07/12/1993	01:13pm	GrandUnion	WheatBread	No

Discovered rules (for John Doe)

(1) Product = LemonJuice => Store = RiteAid (2.4%, 95%)
(2) Product = WheatBread => Store = GrandUnion (3%, 88%)
(3) Product = AppleJuice => CouponUsed = YES (2%, 60%)
(4) TimeOfDay = Morning => DayOfWeek = Saturday (4%, 77%)
(5) TimeOfWeek = Weekend & Product = OrangeJuice => Quantity = Big (2%, 75%)
(6) Product = BabyDiapers => DayOfWeek = Monday (0.8%, 61%)
(7) Product = BabyDiapers & CouponUsed = YES => Quantity = Big (2.5%, 67%)

From Data Mining to Text Mining

- Text Mining, Text Data Mining, Knowledge Discovery from Text, Knowledge Discovery in Textual Data(bases)

"…nontrivial extraction of implicit, previously unknown, and potentially useful information from (large amounts of) textual data"

Text Mining
=
Data Mining (applied to text data)
+
basic linguistics

R. Feldman and I. Dagan, 1995.
Knowledge Discovery in Textual Databases (KDT). In Proceedings of the 1st International Conference on Knowledge Discovery (KDD-95), pp. 112-117, Montreal.

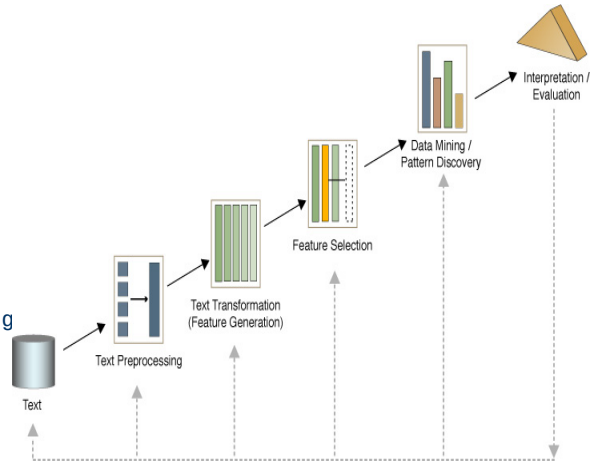
Text Mining

- Discover new knowledge through analysis of text

Database Type	Unstructured
Search Mode	Opportunistic
Atomic entity	Language feature or concept
Example Information Need	"Find the types of food poisoning most often associated with Japanese restaurants"
Example Query	Rank **diseases** found associated with "Japanese restaurants"

Text mining process

- Text preprocessing
  - ✓ Syntactic/Semantic text analysis
- Features Generation
  - ✓ Bag of words
- Features Selection
  - ✓ Simple counting
  - ✓ Statistics
- Text/Data Mining
  - ✓ Classification- Supervised learning
  - ✓ Clustering- Unsupervised learning
- Analyzing results



Text Mining

Discover useful and previously unknown “gems” of information in **large text collections**

Patterns

Trends

Associations

Text Mining@work

Document

I am a Windows NT software engineer seeking a permanent position in a small quiet town 50 - 100 miles from New York City.

I have over nineteen years of experience in all aspects of development of application software, with recent focus on design and implementation of systems involving multi-threading, client/server architecture, and anti-piracy. For the past five years, I have implemented Windows NT services in Visual C++ (in C and C++). I also have designed and implemented multithreaded applications in Java. Before working with Windows NT, I programmed in C under OpenVMS for 5 years.

Filled Template

title: Windows NT software engineer
location: New York City
language: Visual C++, C, C++, Java
platform: Windows NT, OpenVMS
area: multi-threading, client/server, anti-piracy
years of experience: nineteen years

Text Mining@work

Information Extraction

- Input
  - ✓ Natural language documents (newspaper article, email message etc.)
  - ✓ Pre specified entities, templates
- Output
  - ✓ Specific substrings/parts of document which match the template.

Posting from Newsgroup
Telecommunications. Solaris Systems Administrator. 55-60K. Immediate need.

3P is a leading telecommunications firm in need of a energetic individual to fill the following position in the Atlanta office:

SOLARIS SYSTEM ADMINISTRATOR
Salary: 50-60K with full benefits
Location: Atlanta, Georgia no relocation assistance provided

FILLED TEMPLATE
job title: SOLARIS SYSTEM ADMINISTRATOR
salary: 55-60K
city: Atlanta
state: Georgia
platform: SOLARIS
area: Telecommunications

Text Mining@work

- HTML ∈ *language* **and** DHTML ∈ *language* → XML ∈ *language*
- Illustrator ∈ *application* → Flash ∈ *application*
- Dreamweaver 4 ∈ *application* **and** Web Design ∈ *area* → Photoshop 6 ∈ *application*
- MS Excel ∈ *application* → MS Access ∈ *application*
- ODBC ∈ *application* → JSP ∈ *language*
- Perl ∈ *language* **and** HTML ∈ *language* → Linux ∈ *platform*

Text Mining nell'Impresa

"Il processo di estrazione di conoscenza, precedentemente sconosciuta, da fonti testuali (agenzie stampa, transazioni, siti Web, e-mail, forum, mailing list…) utilizzabile per prendere decisioni aziendali"

Permette di organizzare/ categorizzare

- *scoprendo tendenze*
- *apprendendo concetti*

Text Mining nell'Impresa

- Perché è necessario...
  - ✓ scoprire quali sono le opinioni, le idee, le tendenze, i gusti degli utenti (clienti) sta diventando sempre più impegnativo:troppi i dati a disposizione e, troppo rapidi i cambi di tendenza
- ...Le fonti da analizzare
  - ✓ e-mail, newsgroup, forum, mailing list, lettere, articoli, ...
- ...L'obiettivo perseguito
  - ✓ analizzare migliaia di testi in pochi secondi, *raggruppandoli* in funzione del loro *contenuto*, estraendo opinioni, tendenze, idee… *degli autori* (analisi delle lettere di lamentela degli utenti)

Text Mining: aree di ricerca correlate

- Information Retrieval
- Text Categorization
- Information Extraction
- Natural Language Processing
- Data Mining

M. Grobelnik, D. Mladenic, and N. Milic-Frayling, 2000.

"Text Mining as Integration of Several Related Research Areas: Report on KDD'2000 Workshop on Text Mining

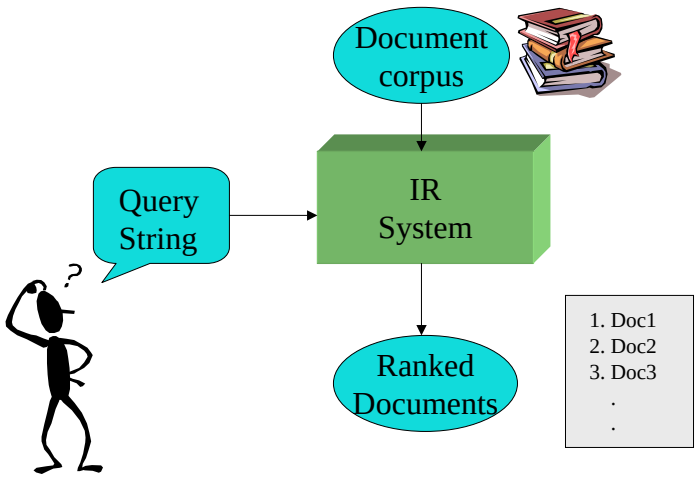
Information Retrieval (IR)

- The indexing and retrieval of textual documents.
- Searching for pages on the World Wide Web is the most recent “killer app.”
- Concerned firstly with retrieving *relevant* documents to a query.
- Concerned secondly with retrieving from *large* sets of documents *efficiently*.

Typical IR Task

- Given:
  - A corpus of textual natural-language documents.
  - A user query in the form of a textual string.
- Find:
  - A ranked set of documents that are relevant to the query.

IR System



Relevance

- Relevance is a subjective judgment and may include:
  - ✓ Being on the proper subject.
  - ✓ Being timely (recent information).
  - ✓ Satisfying the goals of the user and his/her intended use of the information (*information need*).

Keyword Search

- Simplest notion of relevance is that the query string appears verbatim in the document.
- Slightly less strict notion is that the words in the query appear frequently in the document, in any order (*bag of words*).

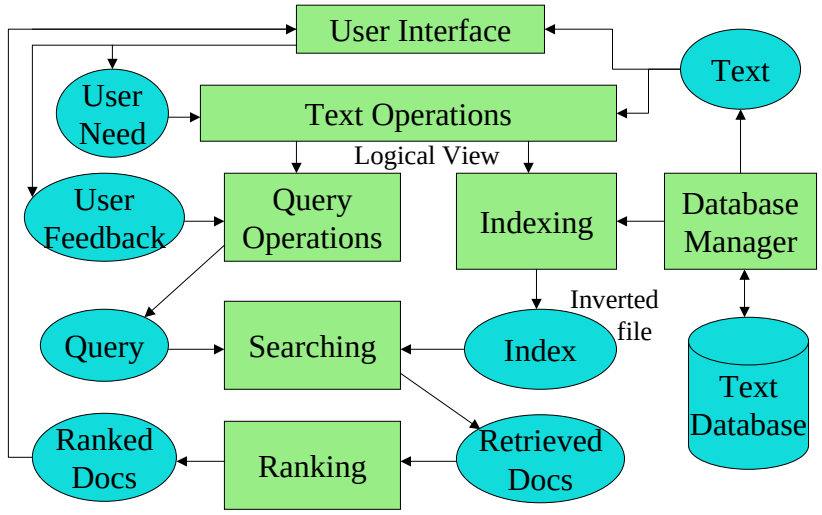
Problems with Keywords

- May not retrieve relevant documents that include *synonymous* terms.
  - ✓ “restaurant” vs. “café”
  - ✓ “PRC” vs. “China”
- May retrieve irrelevant documents that include ambiguous terms (*polysemy*).
  - ✓ “bat” (baseball vs. mammal)
  - ✓ “Apple” (company vs. fruit)

Intelligent IR

- Taking into account the *meaning* of the words used.
- Taking into account the *order* of words in the query.
- Adapting to the user based on direct or indirect feedback (*relevance feedback*): collects feedback, generates new query, repeat retrieval.

IR System Architecture



IR System Components

- ***Text Operations*** forms index words (*tokens*).
  - ✓ Stopword removal
  - ✓ Stemming (reducing words to roots, removing prefix and suffix)
- ***Indexing*** constructs an *inverted index* of word to document pointers.
- ***Searching*** retrieves documents that contain a given query token from the inverted index.
- ***Ranking*** scores all retrieved documents according to a relevance metric. It may also perform *grouping*, i.e. finding commonalities and presenting group of documents.

Intelligent IR?

- Research areas
  - ✓ Natural Language Processing
  - ✓ Machine Learning

Natural Language Processing

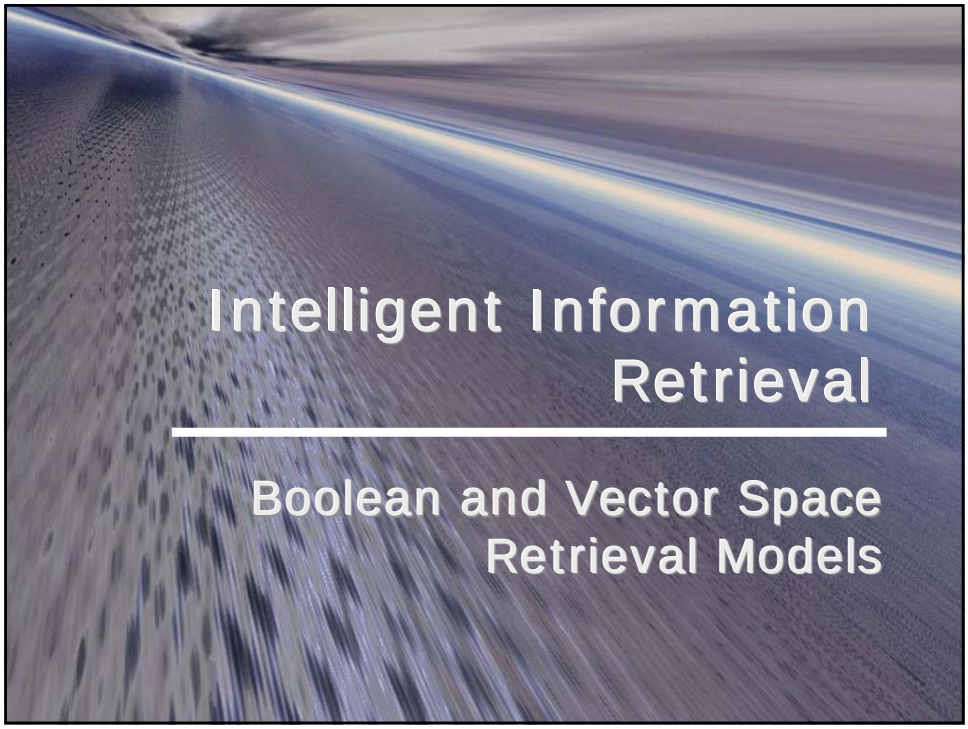
- Focused on the syntactic, semantic, and pragmatic analysis of natural language text and discourse.
- Ability to analyze syntax (phrase structure) and semantics could allow retrieval based on *meaning* rather than keywords.

Natural Lang. Proc: IR Directions

- Methods for determining the sense of an ambiguous word based on context (*word sense disambiguation*).
- Methods for identifying specific pieces of information in a document (*information extraction*).
- Methods for answering specific NL questions from document corpora.

Machine Learning

- Focused on the development of computational systems that improve their performance with experience.
- Automated classification of examples based on learning concepts from labeled training examples (*supervised learning*).
- Automated methods for clustering unlabeled examples into meaningful groups (*unsupervised learning*).



Retrieval Models

- A retrieval model specifies the details of:
  - ✓ Document representation
  - ✓ Query representation
  - ✓ Retrieval function
- Determines a notion of relevance.
- Notion of relevance can be binary or continuous (i.e. *ranked retrieval*).

Classes of Retrieval Models

- Boolean models (set theoretic)
  - ✓ Extended Boolean
- Vector space models (statistical/algebraic)
  - ✓ Generalized VS
  - ✓ Latent Semantic Indexing
- Probabilistic models

Common Preprocessing Steps

- Strip unwanted characters/markup (e.g. HTML tags, punctuation, numbers, etc.).
- Break into *tokens* (keywords) on whitespace.
- Stem tokens to "root" words (retrieval independent of tense, number,...)
  - ✓ computational → comput

  Open problems:
  - ✓ Errors (policies, police → polic)
  - ✓ Loss of context (does → do)
  - ✓ Stem still a word?

Common Preprocessing Steps (cont'd)

References on stemming:

Porter, M., *An algorithm for suffix stripping*, Program, 14(3), 130-137, 1980.

Frakes, W., *Stemming algorithms*, in Frakes, W. & Baeza-Yates, Information Retrieval Data Structures and Algorithms, Prentice-Hall, 1992.

- Remove common stopwords (e.g. a, the, it, etc.).
- Detect common phrases (possibly using a domain specific dictionary).
- Build inverted index (keyword → list of docs containing it).

Boolean Model

- A document is represented as a set of keywords.
- Queries are Boolean expressions of keywords, connected by AND, OR, and NOT, including the use of brackets to indicate scope.
- Output: Document is relevant or not. No partial matches or ranking.

Boolean Retrieval Model

- Popular retrieval model because:
  - ✓ Easy to understand for simple queries.
  - ✓ Clean formalism.
- Boolean models can be extended to include ranking.
- Reasonably efficient implementations possible for normal queries.

Boolean Models – Problems

- Very rigid: AND means all; OR means any.
- Difficult to express complex user requests.
- Difficult to control the number of documents retrieved.
  - ✓ *All* matched documents will be returned.
- Difficult to rank output.
  - ✓ *All* matched documents logically satisfy the query.
- Difficult to perform relevance feedback.
  - ✓ If a document is identified by the user as relevant or irrelevant, how should the query be modified?

Statistical Models

- A document is typically represented by a *bag of words* (unordered words with frequencies).
- Bag = set that allows multiple occurrences of the same element.
- User specifies a set of desired terms with optional weights:
  - ✓ Weighted query terms:
    Q = < database 0.5; text 0.8; information 0.2 >
  - ✓ Unweighted query terms:
    Q = < database; text; information >
  - ✓ No Boolean conditions specified in the query.

Statistical Retrieval

- Retrieval based on *similarity* between query and documents.
- Output documents are ranked according to similarity to query.
- Similarity based on occurrence *frequencies* of keywords in query and document.
- Automatic relevance feedback can be supported:
  - ✓ Terms in relevant documents “added” to query.
  - ✓ Terms in irrelevant documents “subtracted” from query.

Issues for Vector Space Model

- Words or word stems?
- How to determine important words in a document?
  - ✓ Word sense?
  - ✓ Word n-grams (and phrases, idioms,...) → terms
- How to determine the degree of importance of a term within a document and within the entire collection?
- How to determine the degree of similarity between a document and the query?
- In the case of the web, what is a collection and what are the effects of links, formatting information, etc.?

The Vector-Space Model

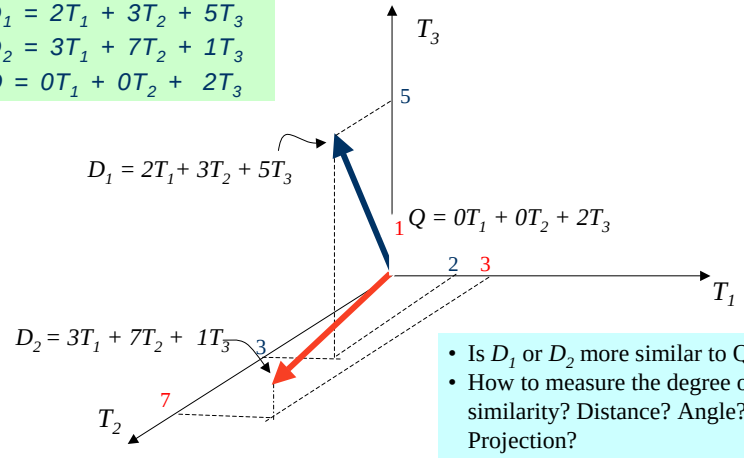
- Assume *t* distinct *terms* remain after preprocessing; call them *index terms* or the *vocabulary*.
- These "orthogonal" (uncorrelated) terms form a vector space.

  Dimension = *t* = |vocabulary|
- Each term, *i*, in a document or query, *j*, is given a real-valued weight, w_{ij}.
- Both documents and queries are expressed as t-dimensional vectors:

  $d_j = (w_{1j}, w_{2j}, ..., w_{tj})$

Graphic Representation

Example:
$D_1 = 2T_1 + 3T_2 + 5T_3$
$D_2 = 3T_1 + 7T_2 + 1T_3$
$Q = 0T_1 + 0T_2 + 2T_3$



- Is D_1 or D_2 more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?

Document Collection

- A collection of n documents can be represented in the vector space model by a term-document matrix.
- An entry in the matrix corresponds to the "weight" of a term in the document; zero means the term has no significance in the document or it simply doesn't exist in the document.

	T_1	T_2	….	T_t
D_1	w_{11}	w_{21}	…	w_{t1}
D_2	w_{12}	w_{22}	…	w_{t2}
:	:	:		:
:	:	:		:
D_n	w_{1n}	w_{2n}	…	w_{tn}

Term Weights: Term Frequency

- More *frequent* terms in a document are more important, i.e. *more* indicative of the topic.
  f_{ij} = frequency of term *i* in document *j*
- May want to normalize *term frequency* (*tf*) across the entire corpus:
  $tf_{ij} = f_{ij} / max\{f_{ij}\}$
- Problems with common words and/or long documents

Term Weights: Inverse Doc. Frequency

- Terms that appear in many *different* documents are *less* indicative of overall topic.
  df_i = document frequency of term *i*
  = number of documents containing term *i*
  idf_i = inverse document frequency of term *i*,
  = $\log_2 (N/ df_i)$
  (N: total number of documents)
- An indication of a term's *discrimination* power.
- idf_i used to dampen the effect relative to *tf*.

TF-IDF Weighting

- A typical combined term importance indicator is *tf-idf weighting*:

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} \log_2 (N/ df_i)$$

- A term occurring frequently in the document but rarely in the rest of the collection is given high weight.
- Many other ways of determining term weights have been proposed.
- Experimentally, *tf-idf* has been found to work well.

Computing TF-IDF: An Example

Given a document containing terms with given frequencies:
A(3), B(2), C(1)
Assume collection contains 10,000 documents and
document frequencies of these terms are:
A(1300), B(600), C(20)
Then:

A: tf = 3/3; idf = $\log_2$(10000/1300) = 2.94; tf-idf = 2.94
B: tf = 2/3; idf = $\log_2$ (10000/600) = 4.06; tf-idf = 2.71
C: tf = 1/3; idf = $\log_2$ (10000/20) = 8.96; tf-idf = 2.98

Query Vector

- Query vector is typically treated as a document and also tf-idf weighted (often terms occur once in a query, then just idf is enough).
- Alternative is for the user to supply weights for the given query terms.
- Stemming and stop words optionally

Similarity Measure

- A similarity measure is a function that computes the *degree of similarity* between two vectors.
- Using a similarity measure between the query and each document:
  - ✓ It is possible to rank the retrieved documents in the order of presumed relevance.
  - ✓ It is possible to enforce a certain threshold so that the size of the retrieved set can be controlled.

Similarity Measure: Inner Product

- Similarity between vectors for the document d_j and query q can be computed as the vector inner product:
- $$sim(d_j,q) = d_j \bullet q = \sum_{i=1}^{t} w_{ij} \cdot w_{iq}$$
  - where w_{ij} is the weight of term i in document j and w_{iq} is the weight of term i in the query
- For binary vectors, the inner product is the number of matched query terms in the document (size of intersection).
- For weighted term vectors, it is the sum of the products of the weights of the matched terms.

Inner Product -- Examples

Binary:

	retrieval	database	architecture	computer	text	management	information
D =	1,	1,	1,	0,	1,	1,	0
Q =	1,	0,	1,	0,	0,	1,	1

sim(D, Q) = 3

Size of vector = size of vocabulary = 7
0 means corresponding term not found in document or query

Weighted:

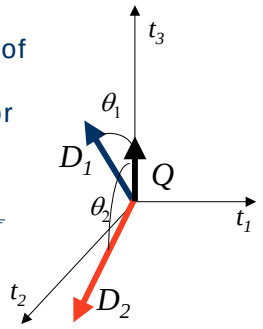
$D_1 = 2T_1 + 3T_2 + 5T_3$ $\quad D_2 = 3T_1 + 7T_2 + 1T_3$
$Q = 0T_1 + 0T_2 + 2T_3$

$sim(D_1, Q) = 2*0 + 3*0 + 5*2 = 10$
$sim(D_2, Q) = 3*0 + 7*0 + 1*2 = 2$

Cosine Similarity Measure

- Cosine similarity measures the cosine of the angle between two vectors.
- Inner product normalized by the vector lengths.

$$\text{CosSim}(\boldsymbol{d}_j, \boldsymbol{q}) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$



$D_1 = 2T_1 + 3T_2 + 5T_3$ CosSim(D_1, Q) = 10 / $\sqrt{(4+9+25)(0+0+4)}$ = 0.81
$D_2 = 3T_1 + 7T_2 + 1T_3$ CosSim(D_2, Q) = 2 / $\sqrt{(9+49+1)(0+0+4)}$ = 0.13
$Q = 0T_1 + 0T_2 + 2T_3$

D_1 is 6 times better than D_2 using cosine similarity but only 5 times better using inner product.

Naïve Implementation

- Convert all documents in collection D to tf-idf weighted vectors, dj, for keyword vocabulary V.
- Convert query to a tf-idf-weighted vector q.
- For each d_j in D do
  Compute score s_j = cosSim(d_j, q)
- Sort documents by decreasing score.
- Present top ranked documents to the user.
- Time complexity: O(|V|·|D|) Bad for large V & D !
  |V| = 10,000; |D| = 100,000; |V|·|D| = 1,000,000,000

Comments on VS Model

- Simple, mathematically based approach.
- Considers both local (*tf*) and global (*idf*) word occurrence frequencies.
- Provides partial matching and ranked results.
- Tends to work quite well in practice despite obvious weaknesses.
- Allows efficient implementation for large document collections.

Problems with VS Model

- Missing semantic information (e.g. word sense).
- Missing syntactic information (e.g. phrase structure, word order, proximity information).
- Assumption of term independence (e.g. ignores synonomy).
- Lacks the control of a Boolean model (e.g., *requiring* a term to appear in a document).
  - Given a two-term query “A B”, may prefer a document containing A frequently but not B, over a document that contains both A and B, but both less frequently.

Intelligent Information Retrieval

Integrazione di conoscenza lessicale: WordNet

WordNet

- Ontologia linguistica che rappresenta in maniera esplicita e formale la conoscenza linguistica umana
- L'idea nasce nel 1985 da un gruppo di linguisti e psicolinguisti dell'università di Princeton
  - ✓ Obiettivo: ricerca concettuale nei dizionari
  - ✓ Risultato: definizione di un database lessicale
  - ✓ Linea di ricerca: memoria lessicale umana
- URL: http://www.cogsci.princeton.edu/~wn/

WordNet

- WordNet è un'ontologia linguistica top-level
- La conoscenza linguistica :
  - ✓ è conoscenza di senso comune
  - ✓ può essere utilizzata in qualsiasi dominio

Utilizzo di WordNet

- Sistemi per Information Retrieval e Text Categorization utilizzano la conoscenza linguistica di WordNet per aggiungere "semantica" al processo di ritrovamento/categorizzazione
  - ✓ Algoritmi di base per l'indicizzazione
  - ✓ Algoritmi avanzati di *word sense disambiguation*

Le quattro categorie lessicali

- La memoria lessicale umana si suddivide in quattro parti ognuna rispettivamente dedicata a: nomi, verbi, aggettivi e avverbi
- Gli ideatori di WordNet, ispirandosi a tale teoria, hanno suddiviso in modo analogo la conoscenza lessicale

Concetto di parola

- PAROLA: un'associazione fra una word form e una word meaning
  - ✓ *word form*: espressione fisica della parola ovvero l'insieme di lettere che la costituisce (stringa)
  - ✓ *word meaning*: concetto lessicale che la word form vuole esprimere ovvero il suo significato sottinteso

WordNet: la matrice lessicale

Word Meanings	Word Forms					
	F_1	F_2	F_3	...	...	F_n
M_1	V(1,1)	V(2,1)				
M_2		V(2,2)	V(3,2)			
M_3						
$M_{...}$						
M_m						V(m,n)

Realizza il mapping tra word form e word meaning

Polysemy & Synonymy

- Una word form è polysemous se ad essa possono essere associate più word meaning
- Due word form sono synonym se ad esse è associata la stessa word meaning

Rappresentazione della conoscenza linguistica

- Lo scopo principale di WordNet è quello di riuscire a trasferire ad un computer tutta la conoscenza linguistica
  - le word form, le word meaning e il mapping fra queste due categorie
- La rappresentazione delle word form, in una forma comprensibile ad un calcolatore, non ha suscitato molti problemi

Rappresentazione della conoscenza linguistica

- Ogni word meaning è rappresentata dall'insieme delle word form che possono essere usate per esprimerla: *synset*
- Un synset associato ad una word form consente all'utente di inferire la semantica della word form in esame purché conosca la semantica di almeno una word form elencata nel synset

Document representation

Journal of Artificial Intelligence Research

JAIR is a referred journal, covering all areas of Artificial Intelligence, which is distributed free of charge over the Internet. Each volume of the journal is also published by Morgan Kaufman…

1	Journal
1	Intelligence
1	Artificial
1	Research

Slot "title"

2	Journal
1	Intelligence
1	Artificial
…	…

Slot "abstract"

Extended Document Representation

Journal of Artificial Intelligence Research

JAIR is a referred journal, covering all areas of Artificial Intelligence, which is distributed free of charge over the Internet. Each volume of the journal is also published by Morgan Kaufman…

Artificial Intelligence

Slot "title"

Why System Evaluation?

- There are many retrieval models/ algorithms/ systems, which one is the best?
- What is the best component for:
  - ✓ Ranking function (dot-product, cosine, ...)
  - ✓ Term selection (stopword removal, stemming...)
  - ✓ Term weighting (TF, TF-IDF,...)
- How far down the ranked list will a user need to look to find some/all relevant documents?

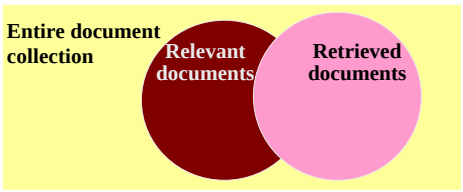
Difficulties in Evaluating IR Systems

- Effectiveness is related to the relevancy of retrieved items.
- Relevancy is not typically binary but continuous.
- Even if relevancy is binary, it can be a difficult judgment to make.
- Relevancy, from a human standpoint, is:
  - ✓ Subjective: Depends upon a specific user's judgment.
  - ✓ Situational: Relates to user's current needs.
  - ✓ Cognitive: Depends on human perception and behavior.
  - ✓ Dynamic: Changes over time.

Human Labeled Corpora

- Start with a corpus of documents.
- Collect a set of queries for this corpus.
- Have one or more human experts exhaustively label the relevant documents for each query.
- Typically assumes binary relevance judgments.
- Requires considerable human effort for large document/query corpora.

Precision and Recall



	retrieved	not retrieved
irrelevant	retrieved & irrelevant	Not retrieved & irrelevant
relevant	retrieved & relevant	not retrieved but relevant

$$recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents}$$

$$precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ documents\ retrieved}$$

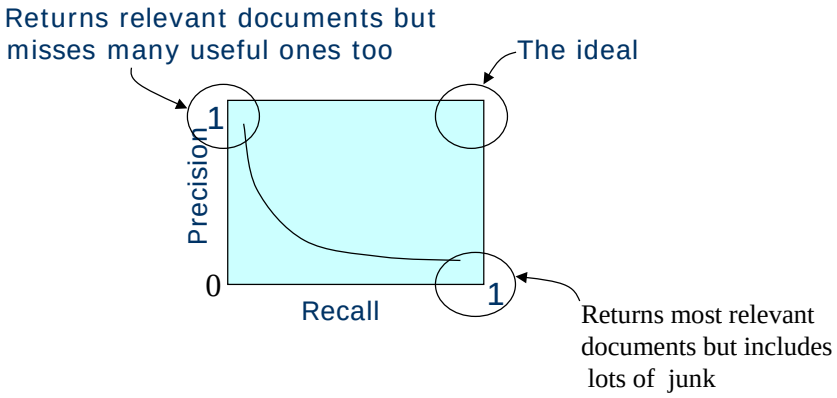
Precision and Recall

- Precision
  - ✓ The ability to retrieve top-ranked documents that are mostly relevant.
- Recall
  - ✓ The ability of the search to find ***all*** of the relevant items in the corpus.

Determining Recall is Difficult

- Total number of relevant items is sometimes not available:
  - Sample across the database and perform relevance judgment on these items.
  - Apply different retrieval algorithms to the same database for the same query. The aggregate of relevant items is taken as the total relevant set.

Recall vs. Precision



Trade-off between Recall and Precision

Computing Recall/Precision Points

- For a given query, produce the ranked list of retrievals.
- Adjusting a threshold on this ranked list produces different sets of retrieved documents, and therefore different recall/precision measures.
- Mark each document in the ranked list that is relevant according to the gold standard.
- Compute a recall/precision pair for each position in the ranked list that contains a relevant document.

Computing Recall/Precision

n	doc #	relevant
1	588	x
2	589	x
3	576	
4	590	x
5	986	
6	592	x
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	x
14	990	

Let total # of relevant docs = 6
Check each new recall point:

R=1/6=0.167; P=1/1=1

R=2/6=0.333; P=2/2=1

R=3/6=0.5; P=3/4=0.75

R=4/6=0.667; P=4/6=0.667

Missing one relevant document. Never reach 100% recall

R=5/6=0.833; P=5/13=0.38

R- Precision

- Precision at the R-th position in the ranking of results for a query that has R relevant documents.

n	doc #	relevant
1	588	x
2	589	x
3	576	
4	590	x
5	986	
6	592	x
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	x
14	990	

R = # of relevant docs = 6

R-Precision = 4/6 = 0.67

F-Measure

- One measure of performance that takes into account both recall and precision.
- Harmonic mean of recall and precision:

$$F = \frac{2PR}{P+R} = \frac{2}{\frac{1}{R}+\frac{1}{P}}$$

- Compared to arithmetic mean, both need to be high for harmonic mean to be high.

$F_ß$ Measure (parameterized F Measure)

- A variant of F measure that allows weighting emphasis on precision over recall:

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R} = \frac{(1+\beta^2)}{\frac{\beta^2}{R}+\frac{1}{P}}$$

- Value of β controls trade-off:
  - ✓ β = 1: Equally weight precision and recall ($F_ß$=F)
  - ✓ β > 1: Weight recall more
  - ✓ β < 1: Weight precision more
  - ✓ β = 0: $F_ß$ =P
- E Measure = 1- $F_ß$

References

- Baeza-Yates, R.A., Ribeiro-Neto, B.A., "Modern Information Retrieval", ACM Press/Addison-Wesley, 1999.
- R. Feldman and I. Dagan, Knowledge Discovery in Textual Databases (KDT). *Proc. of the 1st Int. Conf. on Knowledge Discovery (KDD-95)*, pp. 112-117, Montreal, 1995.
- Frakes, W., Stemming algorithms, in Frakes, W. & Baeza-Yates, *Information Retrieval Data Structures and Algorithms*, Prentice-Hall, 1992.
- M. Grobelnik, D. Mladenic, and N. Milic-Frayling, *Text Mining as Integration of Several Related Research Areas: Report on KDD'2000 Workshop on Text Mining*, 2000.
- M. Pazzani, *Machine Learning and Information Filtering on the Internet*, IJCAI-97 Tutorial, Nagoya, Japan, Aug 1997.
- Porter, M., An algorithm for suffix stripping, *Program*, 14(3):130-137, 1980.
- Salton, G., & McGill, M. J. "Introduction to Modern Information Retrieval", McGraw-Hill, 1983.
- Salton, G., & Buckley, C., Term weighting approaches in automatic text retrieval, *Information Processing and Management*, 24(5):513-523, 1988.
- Salton, G., & Buckley, C., Improving retrieval performance by relevance feedback, *Journal of the American Society for Information Science*, 41:288-297, 1990.
- vanRijsbergen, C.J., "Information Retrieval", Butterworth & Co., Boston, MA, 1979.