INGENIERÍA INDUSTRIAL
UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Chapter 3: Web Content Mining

Professors:	Juan D. Velásquez	http://wi.dii.uchile.cl/
	Gaspar Pizarro V.	@juandvelasquez

Section 3.2

The Classic Text Mining procces applied to Web pages

The Web Text

- In order to analyze we **need to process** before to use it.
  - Document free text (without tags)
  - Stop-word filtering (to explain later)
  - Stemming algorithm (to explain later)
- All these procedure are performed to have a **clean list of words** that represent a web page.
- We want to transform these lists of words into objects we can manipulate using math…

The Vector Space Model

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

The word page vector (wp)

- Each web page can be considered as a **document text** with tags.
- Applying filters, the web page is transformed to the **feature vector**.
- Let $P = \{p_1, \ldots, p_Q\}$ be the **set of *Q* pages** in a web site.
- The i-th page is represented by

$$wp^i = \{wp^i_1, \ldots, wp^i_R\} \in WP$$

- with *R* the number of words after a **stop word** and **stemming process** and WP the set of feature vectors.

The word page vector (wp) (2)

- Meaning of the ***k***-component (wp^i_k) of the feature vector: "The importance of the word ***k*** on the page ***i***"

- With this model, we have transformed the bags of words into vectors and matrices, so we now:
  - Have a **numeric representation** of text
  - **Can compare** 2 pages (documents)
  - Can use a more **complex battery of mathematical tools** for text analysis and mining.
  - Can First (approximate) approach to **representing the meaning of a page** by list of words.

Building Vector Space Model

- From different web page content, special attention receive the **free text**.
- For the moment, a searching is performed by using key words.
- It is necessary to *represent the text information in a* ***feature vector***, before to apply a mining process.
- The representation must consider that *the words in the web page don't have the same importance.*

Stages of the process of building the Vector Space Model

I. **Parsing** the web page content
II. Deleting unnecessary content
III. Identifying the text semantic: **Stemming**
IV. **Calculating** the feature vector
V. Data mining **algorithm** application:
    I. clustering and similarity measure.

I. Parsing the content: Tokenizing

- **Extract text content**:
  - individualizing each word contained in the document.
- A web document is based on **HTML tags**
- The usual procedure is to extract all the free text word *avoiding all the HTML tag*.
- **Filtering**:
  - Also commonly removing **stop word** like :
    - “the”, “a”, “by”, “he”, “she”, “behind”, “above”, “below”, …
- **Result:**
  - **A raw list of word for each page.**

II. Deleting Stopwords

- A **full list** of them (in English):
  http://dev.mysql.com/doc/refman/5.0/en/fulltext-stopwords.html
- **The Semantic**:
  - the study of the meaning in a communication process.
- In vector space model:
  - We need to identify the **importance of a word in a text**, from the point of view of the **semantic**.
- Our first approximation: *"Stop Word doesn't contribute to the semantic of a text."*
- **It is an approximation**.
- **We cannot capture the semantic** of *"the main course was explained in the thai food course ".*

III. Identifying the semantics: Stemming

- There are words that have **“similar meaning”:** *connect, connected, connecting, …*
- It is necessary to associate a **unique identifier** of the semantic content for them.
- **Word stemming**:
  - A way to generate word with unique semantic.
- {connect, connected, connecting} -> “connect”

III. Identifying the semantics: Stemming (2)

- First work on 1968 Lovins.
- **Martin Porter** http://tartarus.org/martin/
- **Stemming**:
  - The process for removing the commoner morphological and inflexional endings from word.
- This process is widely used in *Information Retrieval Process Systems.*
- This process has the intention **to extract the semantic root** of word in a document, in order to have a more *simpler description* of the semantic of the text content.
- Usually the process works in language like **English**, others like **Arabic**, **Hebrew** are more difficult to stem.

III. Identifying the semantics: Stemming (3)

THE PORTER ALGORITHM

1. Take the next **word** on the text
2. Determine if it has suffixes, like: -ED, -ING, -ION, -IONS, ... and others
3. Lookup in the exception rule list if the **word** is present and then apply the rule
   - Ex: ran->run
4. If not then cut the suffix and return the remaining part
   1. Ex: connections -> connect
5. Insert the **new word** on a list and return to the ***step 1*** if another word remain on the text; if not then finish and return the list of processed **new word**

III. Identifying the semantics: Stemming (4)

THE PORTER ALGORITHM

- From the Porter page:
  http://tartarus.org/martin/PorterStemmer/index.html
- **Snowball** Library:
  - JAVA available
  - More robust support
  - Other languages supported than **English**, like **Spanish.**

  http://snowball.tartarus.org/download.php

IV. Calculating the Word Page Vector

- We have a ***clean list of stemmed word*** for each page.
- *¿How can we calculate the numeric importance of a word on the page?*
  - **Binary measure**:
    - 1 if the word k is present on i, 0 if not.
  - **Frequency measure**:
    - the *relative frequency* of the word k on the page i vs.. all the pages.
  - **Other measures**:
    - Next page

IV. Calculating the Word Page Vector (2)

- Its **vector representation** would be a matrix of *RxQ*.
- *Q* is the number of pages in the web site and *R* is the number of different words in *P*.

	Word	1	2	…	Q
1	advise	1	0	…	1
2	business	0	1	…	0
.	…	.	.	…	.
.	…	.	.	…	.
.	…	.	.	…	.
.	…	.	.	…	.
.	…	.	.	…	.
R	zambia	1	0	…	0

IV. Calculating the Word Page Vector (3)

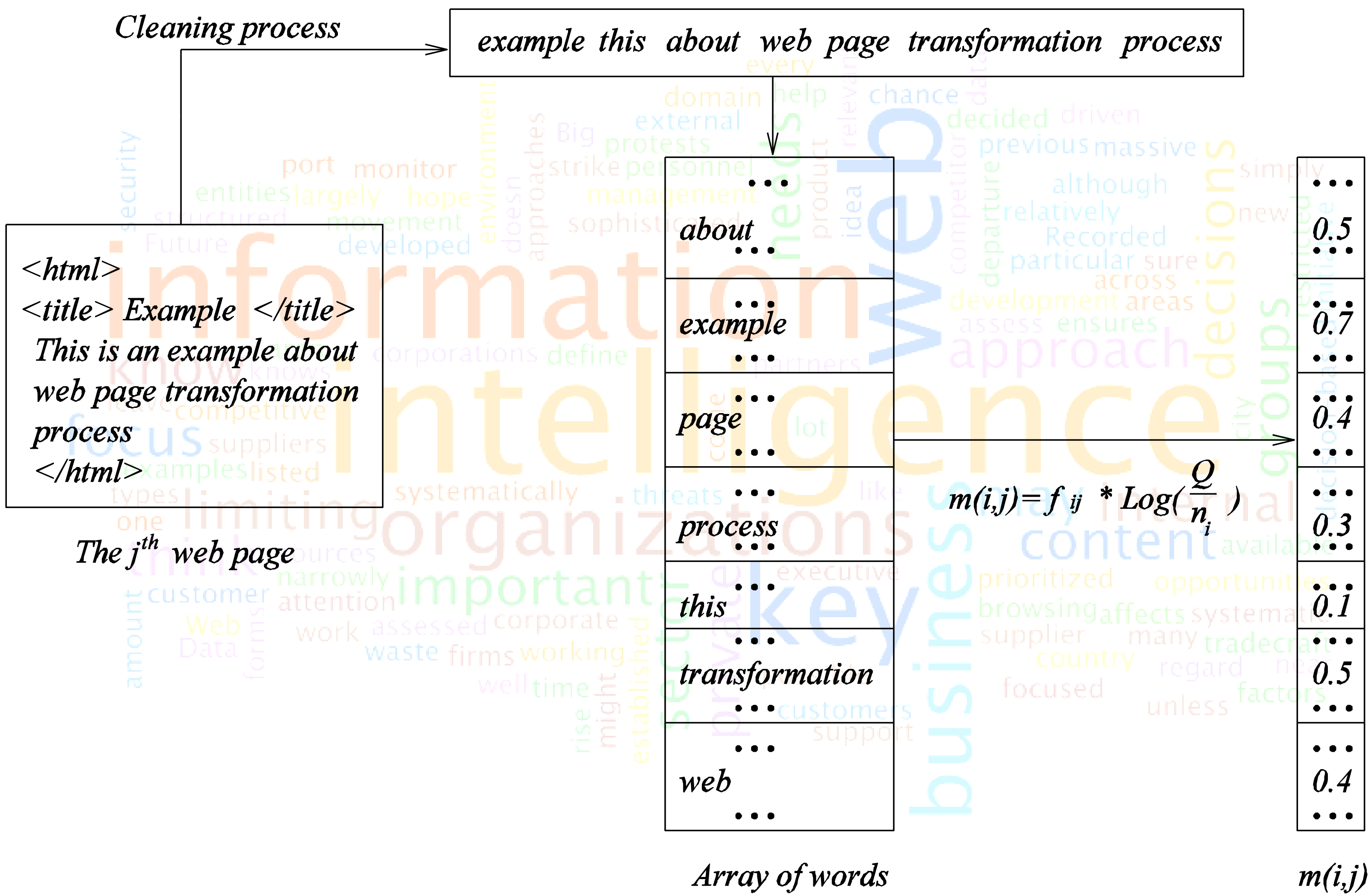
- The model associates a **weight to each word** in the page, based on its *frequency* in the whole web site.
- Let n_i the number of pages with the word i and Q the amount of pages, a simple estimation of the relevance of a word is:

$$wp^i{}_j = n_i / Q$$

- The ***inverse document frequency*** can be used like a weight. $wp^i{}_j = IDF = \log(Q / n_i)$
- *A variation of the last expression is known as TF*IDF,* where f_{ij} the number of occurrences of word i in the document j: $wp^i{}_j = TF * IDF = f_{ij} * \log(Q / n_i)$

IV. Calculating the Word Page Vector (4)

EXAMPLE



IV. Calculating the Word Page Vector (5)

DIFFERENT APPROACHES

- Based on the **TF*IDF weights**:

  $$wp_{ij} = f(i,j) * \log(Q/n_i)$$

- A more **parameterized approach**:

  $$wp_{ij} = f(i,j) *(1+ sw_i)* \log(Q/n_i)$$

- Where sw_i is an **additional weight** that for the i-th word.
- In this way, the vector sw_i allows to include **semantic information** about *special word* in the page like tagged word in HTML (bold, italic, titles,…).

IV. Calculating the Word Page Vector (6)

DIFFERENT APPROACHES

- Another suggestion is

  $$wp_{iq} = (0.5 + [0.5 * freq(i,q) / max(freq(l,q))]) * log(N / n_i)$$

- This model is very good in practice:
  - TF*IDF works well with **general collections**
  - **Simple and fast to compute**
  - *Vector model* is usually as good as the *known ranking alternatives*
- **Why?** : These results are validated by **empirical** experiment.

IV. Calculating the Word Page Vector (7)

DIFFERENT APPROACHES

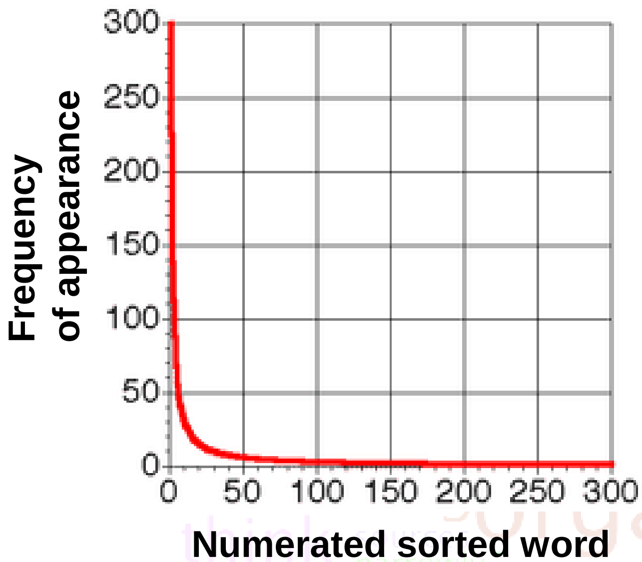
- From a 1945 study on **free text** in a document repository.
- Shown that the graph **Log(Frecuency of use of a word) vs. -Log(Number of Word)** is **Linear!!**
- This rule was verified on **several other document repository**, even in web text.
- That mean that **different word distribution** on a text follows a power law:

$$P(n)\ \mu\ n^{-b}$$

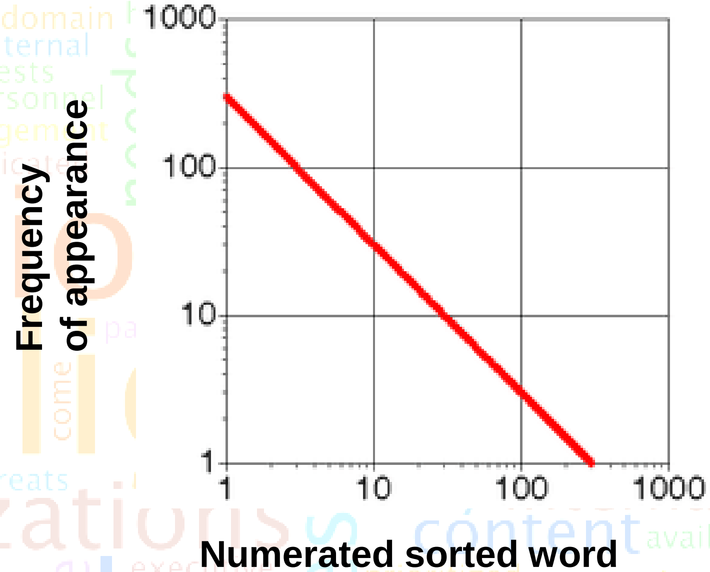
IV. Calculating the Word Page Vector (8)

DIFFERENT APPROACHES

Linear scale



Log-log scale



- If the **text frequencies follows a power laws**, then the weight measure like IDF retrieve us the a *narrow approach to the most important* **(=most probable**) word in a linear way.

V. Data Mining

- Now we have <u>vectors that represent pages and word importance over them</u>.

$$wp^i = (wp^i_1, \ldots, wp^i_R) \in WP$$

- We **have to process** this data
- Data mining techniques applies
- … But we need to *define a way to compare them.*

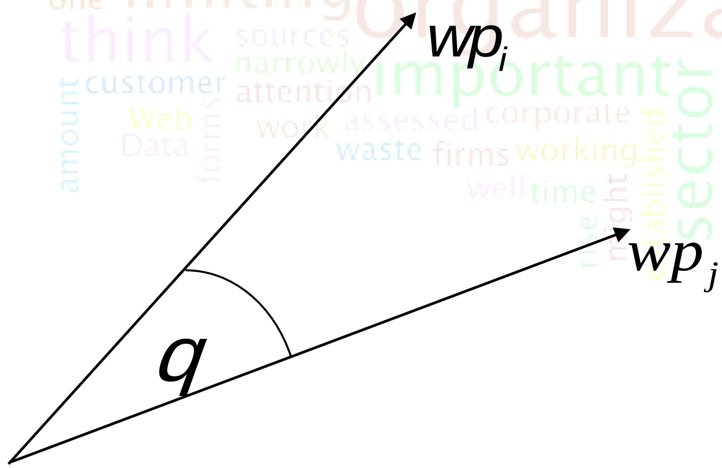
V. Data Mining (2)

THE SIMILARITY MEASURE

$$wp = (wp_{ij}) = f_{ij} * \log(\frac{Q}{n_i})$$

$$wp_i \rightarrow (wp_{1i}, \ldots, wp_{Ri}) \quad wp_j \rightarrow (wp_{1j}, \ldots, wp_{Rj})$$

$$dp(wp_i, wp_j) = \cos\theta = \frac{\sum_{k=1}^{R} wp_{ki} wp_{kj}}{\sqrt{\sum_{k=1}^{R} (wp_{ki})^2} \sqrt{\sum_{k=1}^{R} (wp_{kj})^2}}$$



V. Data Mining (3)

THE SIMILARITY MEASURE

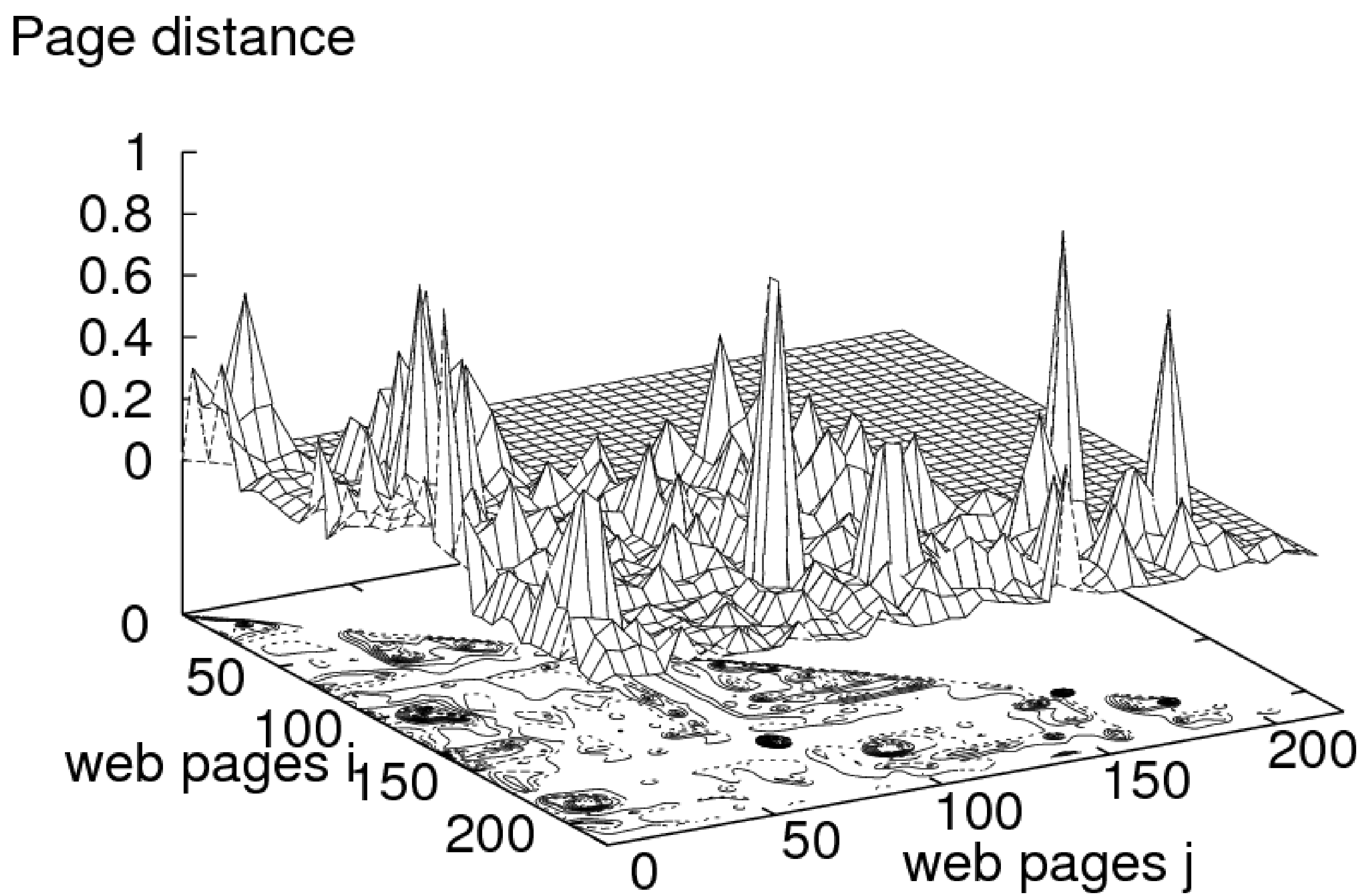
- Any method for **grouping needs to have an understanding for how similar observation are to each others** (*clustering*).
- *You don't need to have triangular inequality property*.
- In the case of the cosine measure, we have the benefits that is scale invariant. The **Euclidean distance** function doesn't have this property!! That is:

$$dp(wp_i, wp_j) = dp(l\,wp_i, l\,wp_j)$$

- *We need this property because the **units** of the wp values are NOT important for the TEXT PROBLEM.*

V. Data Mining (4)

THE SIMILARITY MEASURE



Section 3.3

Web Text Mining: Some special aspects and issues

Web Text Mining

- With the word vectors we have given a **numeric meaning to Web pages**.
- *With a similarity measure we could compare word vectors.*
- Then, our data mining algorithms will use **word vectors** as **features.**
- In this section, we'll explore some aspects of the problem:
  - To label or not to label (Supervised/Unsupervised)
  - *Which algorithm to use?, How to compare the results?*
  - Business Applications

Supervised v/s Unsupervised

- Supervised algorithm:
  - Like regression
  - Better adjustment
  - Overfitting issues
  - Be careful with the training set (known labels)
    - Validation set test for adjustment of parameters
    - Test set to measure the quality of the adjustment.
    - Feature selection avoiding curse of dimensionality.
  - Allow to have risk minimization: using the parameter of the model adjusting to the minimal risk or cost function results.

Supervised v/s Unsupervised (2)

- Unsupervised Algorithms:
  - There is no training set with known classification
  - It really can be used to discover hidden information
  - Useful when data are to large in order to examine and label, like surveys.
  - Need human expert verification of the results that sometime could be noisy.
  - Once the expert confirms a correct labeling, we have found natural documents aggregation of data.

Supervised v/s Unsupervised (3)

- In the text mining context:
  - Not all the times we have a clear labeling of the problem. Most of the time the labels need to be discover.
  - When we have some classification of document the supervised method are more precise than the others. But if we have few training examples underfitting issues could be important.

Choosing the right algorithm
STATISTICAL ACCURACY

- Experimentation: A general automatic solution to all problems doesn’t exist. YOU NEED TO EXPERIMENT WITH THE ALGORITHM FIRST.
- Each particular algorithm have pro/cons issues.
- Try to use a “diverse” set of algorithm, and your results will be statistically credible.
- There are meta-algorithm that perform better than each individual one:
  - Bagging or Bootstrap Aggregation
  - Boosting
  - Co-training

Choosing the right algorithm (2)

META ALGORITHMS

- **Bagging or Bootstrap aggregating**:
  - Given a Training Set T, we select random subsets $S_i \subseteq T$, $i=1 \ldots N$
  - For each S_i subset we train N models
  - The final model is the average of the output of the N model. For classification the average correspond to the "majority voting".
- Bagging Properties:
  - Improve classification an regression accuracy
  - Reduce variance
  - Help Avoiding Over fitting
  - Doesn't work if the training set is small.

Choosing the right algorithm (2)

META ALGORITHMS

BAGGING

Training phase

1. Initialize the parameters
   - $\mathcal{D} = \emptyset$, the ensemble.
   - L, the number of classifiers to train.
2. For $k = 1, \ldots, L$
   - Take a bootstrap sample S_k from $\mathbf{Z}$.
   - Build a classifier D_k using S_k as the training set.
   - Add the classifier to the current ensemble, $\mathcal{D} = \mathcal{D} \cup D_k$.
3. Return $\mathcal{D}$.

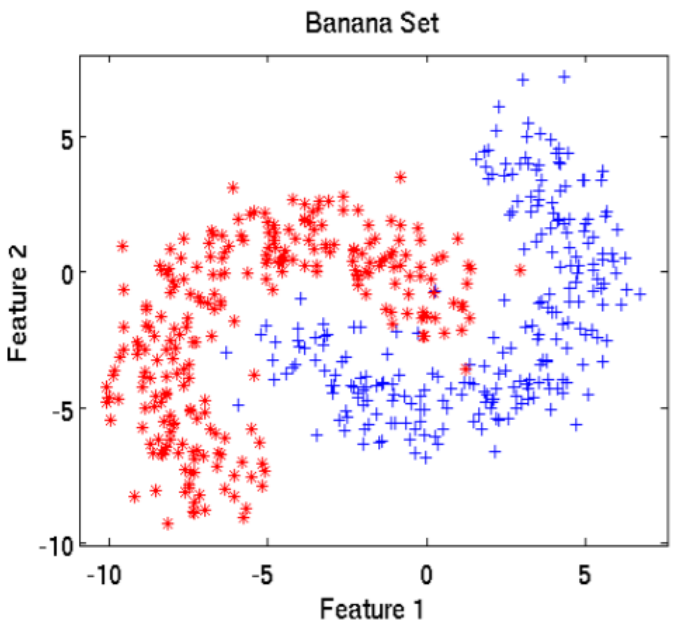
Classification phase

4. Run $D_1, \ldots, D_L$ on the input $\mathbf{x}$.
5. The class with the maximum number of votes is chosen as the label for $\mathbf{x}$.

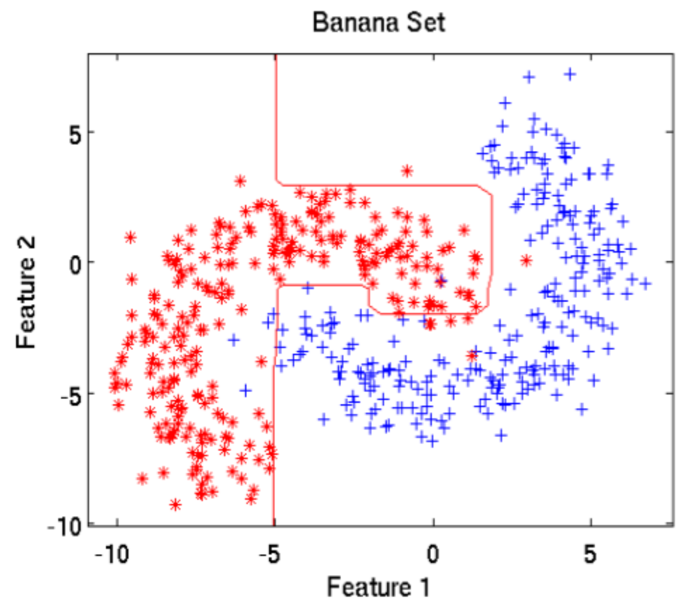
Choosing the right algorithm (2)

META ALGORITHMS

Example



Training data



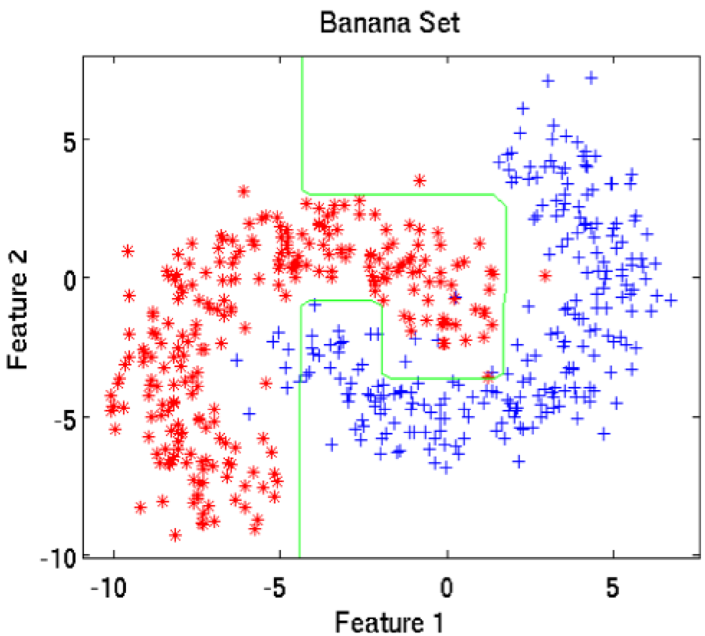
Decision boundary produced by one tree

Bagging: Decision Tree

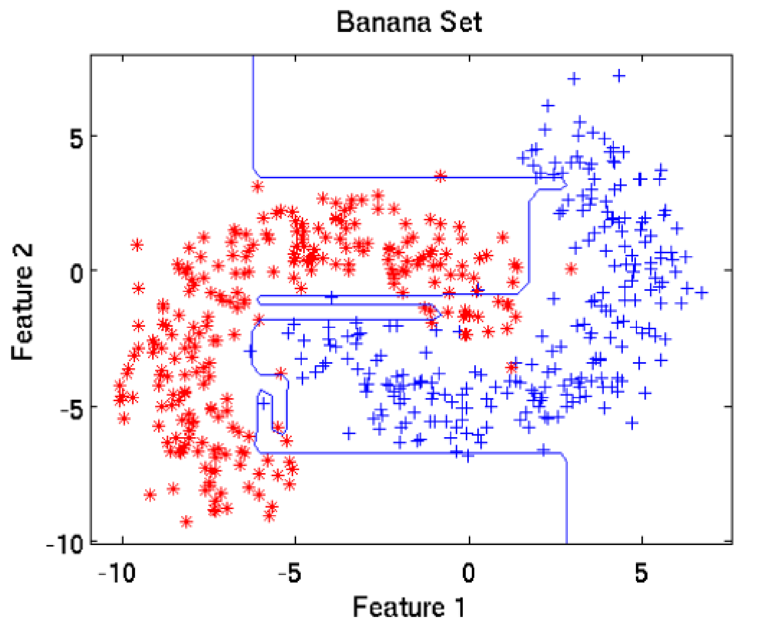
Choosing the right algorithm (2)

META ALGORITHMS

Example



Decision boundary produced by a second tree

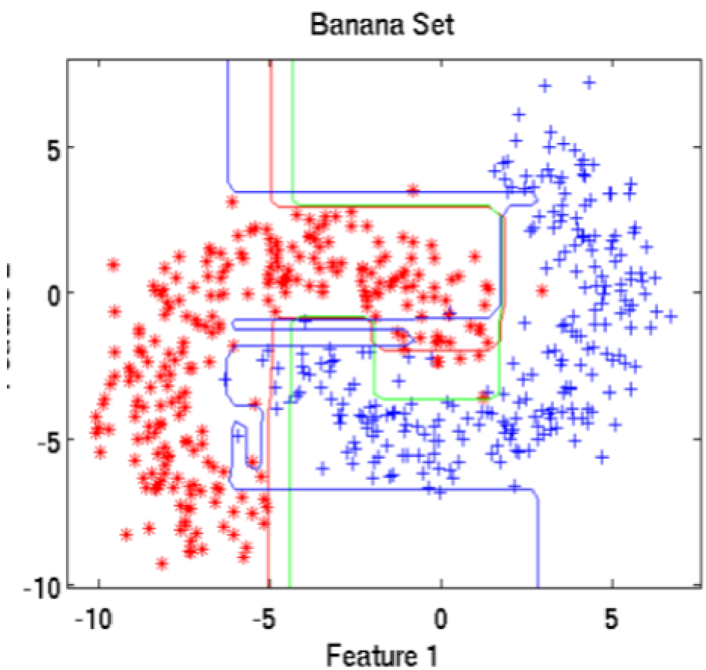


Decision boundary produced by a third tree

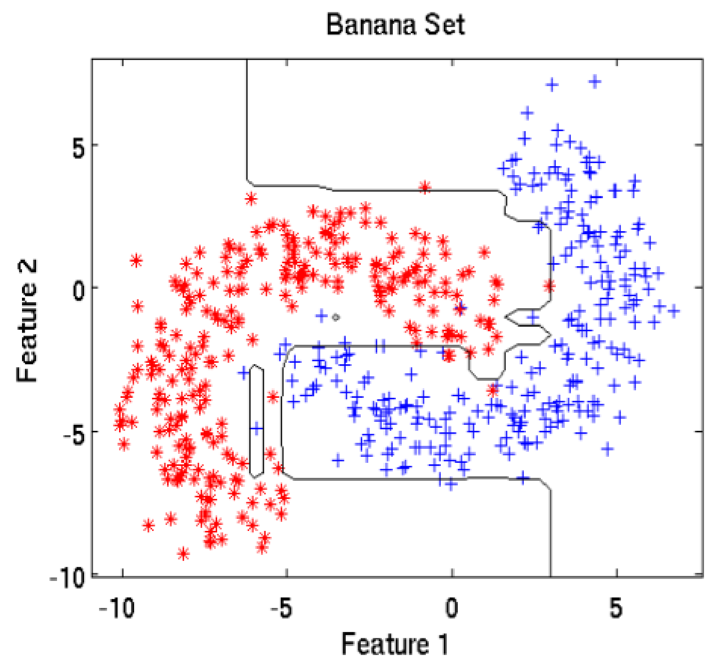
Bagging: Decision Tree

Choosing the right algorithm (2)
META ALGORITHMS

Example



Three trees and final boundary overlaid



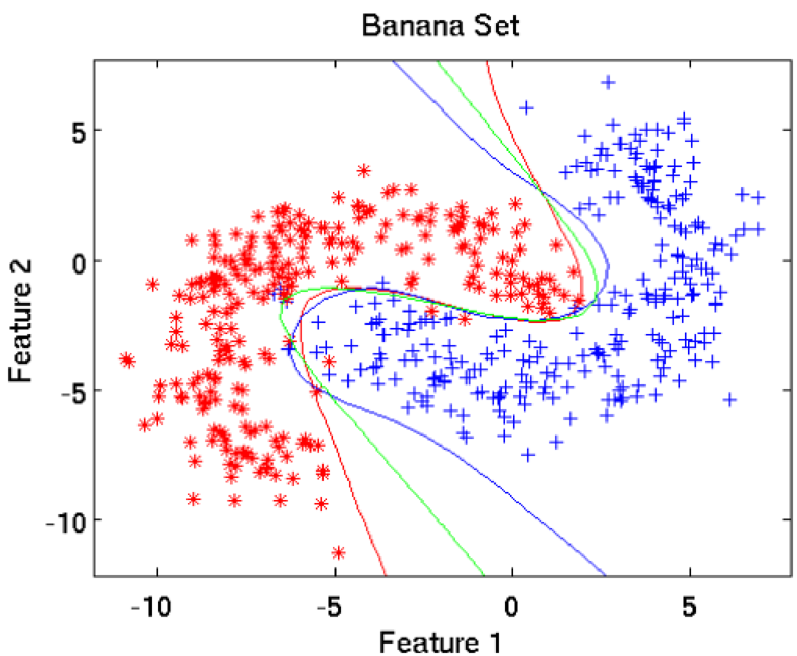
Final result from bagging all trees.

Bagging: Decision Tree

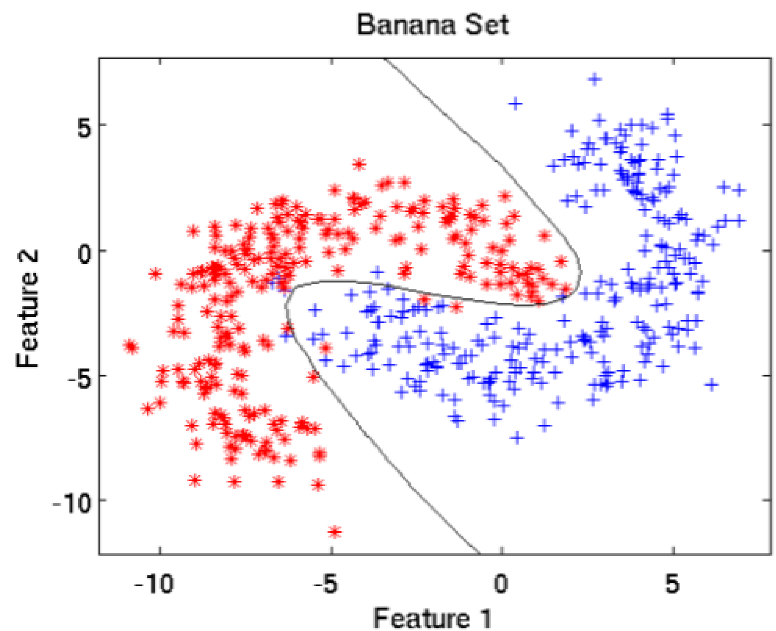
Choosing the right algorithm (2)

META ALGORITHMS

Example



Three neural nets generated with default settings [bpxnc]



Final output from bagging 10 neural nets

Bagging: Neural Net

Choosing the right algorithm (2)

META ALGORITHMS

Why does bagging work ?

- Main reason for error in learning is due to noise ,bias and variance.
- Noise is error by the target function
- Bias is where the algorithm can not learn the target.
- Variance comes from the sampling, and how it affects the learning algorithm
- Does bagging minimizes these errors ?
  - Yes
- Averaging over bootstrap samples can reduce error from variance especially in case of unstable classifiers

Choosing the right algorithm (3)

META ALGORITHMS

- **Boosting**:
  - A technique for combining multiple base classifiers whose combined performance is significantly better than that of any of the base classifiers.
  - Sequential training of weak learners
    - Each base classifier is trained on data that is weighted based on the performance of the previous classifier
  - Each classifier votes to obtain a final outcome

Choosing the right algorithm (3)

META ALGORITHMS

- **Boosting**:
  - Having a set of M different algorithm for data mining.
  - The output result of each algorithm is averaged to produce the final result. In the case of classification the **averaging** is by "majority voting".
- Boosting Properties:
  - Same than bagging.
  - Works also with small training set.
  - The result is always better than individual algorithm approach.
  - The way to average (or combine) the algorithm could be parametric. Like linear combination that call LPBoost, AdaBoost (Adaptive Boost) where the final result is found iterating over the best averaging result.

HEDGE (β)

Given:

- $\mathcal{D} = \{D_1, \ldots, D_L\}$: the classifier ensemble (L strategies)
- $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$: the data set ($N$ trials).

1. Initialize the parameters
   - Pick $\beta \in [0, 1]$.
   - Set the weights $\mathbf{w}^1 = [w_1, \ldots, w_L]$, $w_i^1 \in [0, 1]$, $\sum_{i=1}^{L} w_i^1 = 1$. (Usually $w_i^1 = \frac{1}{L}$).
   - Set $\Lambda = 0$ (the cumulative loss).
   - Set $\lambda_i = 0$, $i = 1, \ldots, L$ (the individual losses).
2. For every $\mathbf{z}_j$, $j = 1, \ldots, N$,
   - Calculate the distribution by

$$p_i^j = \frac{w_i^j}{\sum_{k=1}^{L} w_k^j}, \quad i = 1, \ldots, L. \tag{7.5}$$

   - Find the L individual losses. ($l_i^j = 1$ if D_i misclassifies $\mathbf{z}_j$ and $l_i^j = 0$ if D_i classifies $\mathbf{z}_j$ correctly, $i = 1, \ldots, L$).
   - Update the cumulative loss

$$\Lambda \leftarrow \Lambda + \sum_{i=1}^{L} p_i^j l_i^j \tag{7.6}$$

   - Update the individual losses

$$\lambda_i \leftarrow \lambda_i + l_i^j. \tag{7.7}$$

   - Update the weights

$$w_i^{j+1} = w_i^j \beta^{l_i^j}. \tag{7.8}$$

3. Calculate the return Λ, λ_i, and p_i^{N+1}, $i = 1, \ldots, L$.

Comparing the performance of different algorithms on a problem

- Now we have several algorithmic methods for machine learning.
  - **SVM**
  - **NB**
  - **NN**
  - **KNN**
  - **Bagging or Bootstrap aggregating:**
  - **Boosting**

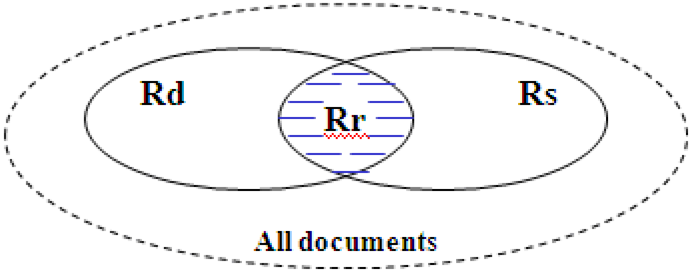
- The problem now is to find a methodology to compare the performance of them in an specific example.

Performance Measures

- **Training Set:** The set from the classifier is constructed.
- **Test Set:** The set from we measure the quality of the classifiers.
- **Breakeven point:** Threshold a confidence value for accepting the declaration of the class label.
- Important values:
  - Rd = Relevant documents
  - Rr = Retrieved relevant documents
  - Rs = Response set

Performance Measures (2)

- R: Recall number = Rr/Rd
- P: Precision = Rr/Rs
- E: Error= TI/N
- Example: If we have 2 classes



	YES is correct	NO is correct
Assigned YES	a	b
Assigned NO	c	d

- R=a/(a+c)
- P=a/(a+b)
- E=(b+c)/N ; of course N=a+b+c+d
- A = (a+d)/N

Performance Measures (3)

- We interpret R as the probability of that a document in the class YES is classified in this class. P is the probability that the document classified in the class YES truly belong to it.
- => We want that both probability be the same. Then The threshold Θ value that we want is the one that do R=P.
- Θ: Is interpreted as the MEASURE of performance of the model.
- This performance measure is accepted in international papers as standard.

Performance Measures (4)

- For more than 2 class label, we use MICROAVERAGING that the previous calculus are performed on each label separately.
- From this we obtain several threshold that equal P and R for each column. We take as the performance value the Minimum Θ^* (worst case) of them.
- Θ^* : The performance measure of the algorithm.

Performance Measures

- Exercise 1: Calculations
  - Assume the following
    - A database contains 80 records on a particular topic.
    - A search was conducted on that topic and 60 records were retrieved
    - Of the 60 records retrieved, 45 were relevant.
    - Calculate the precision and recall scores for the search.

Performance Measures

- Solution
  - A = The number of relevant records retrieved (Rr)
  - B= The number of relevant records not retrieved.
  - C = the number of irrelevant records retrieved.

- In this excersise A=45, B= 35 (80-45) and C =15 (60-45)
- Recall= (45/(45+35))*100 => 56%
- Precision = (45/45+15))*100 => 75%

Business Applications

- Decisions support in CRM:
  - Customer **text complain** analysis
  - The correlation between the **number of satisfied customer and text from them** (emails, messages, etc., …)
- Personalization's in ecommerce
  - Suggestions based on text personal information, messages, emails, text complain.

Business Applications (2)

- Bank customer messages (or email) repository.
  - **Analysis of customers requirement** (urgencies, request, insult, ...)
  - Bank management need "to know" what are the **principal problems on the business**.
  - **Anticipating** problems, **retaining** customers.
  - Allow to **modify the site** *in order to cover new and demanding systems.*

Business Applications (3)

- Online Movies recommender system like http://www.netflix.com/
- **Based on your personal history** and **personal text info**, the system recommend a movie.
- A prize of **1 million US$** for the *best algorithm for recommendation.*
- http://www.netflixprize.com/
- In September 21, 2009 Netflix awarded the $1M Grand Prize to team “BellKor’s Pragmatic Chaos”.

Netflix Prize

Section 3.4

Natural Language Processing Tools for Text Mining

Natural Language Processing (NLP)

- A field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages.
- NLP is related to the area of human–computer interaction.
- Many challenges in NLP involve natural language understanding: enabling computers to derive meaning from human or natural language input.

Part-of-Speech Tagging

- ***Part-of-speech tagging, or POS tagging, is the process that aims to mark up each word in a corpus as corresponding to a particular part of speech***, based on both its definition and its relationship with adjacent or related words in a sentence or paragraph.
- POS tagging has been directly related with the elaboration of Linguistic Corpora. The first tagging process was performed manually during the 60's using the Brown Corpus, one of the biggest corpora of English for computer analysis. The tagging task, which lasted for several years, finished with the development of a program that automatized the process.
- The program was continually improved during the following years and by the late 70's, the algorithm was nearly perfect.

Part-of-Speech Tagging (2)

- Both supervised and unsupervised methods have been proposed, but the first ones are most widely used. There are two main approaches:
  - **Stochastic Methods:** Taking the work with the Brown Corpus as a basis, many statistical approaches have been developed. These techniques include, for instance, the use of **Hidden Markov Models** (or HMMs), which involve counting cases and making a table of the probabilities of certain sequences, and dynamic programming algorithms, which try to solve the same problem in less time.
  - **Rule-based Methods:** Basically, a technique proposed by Eric Brill in his Ph.D. thesis in 1993. This technique learns a set of **patterns** and then applies those patterns rather than optimizing a statistical quantity.

Part-of-Speech Tagging (3)

- In POS tagging, a special issue is determining the *tag set*: the annotation system that will be used to mark each possible part of speech.
- POS tagging work has been done in a variety of languages, and the set of POS tags used varies greatly with each one.
- The number of tags will depend on the purpose at hand. In the case of automatic tagging, it is obviously better to have smaller tag sets.
- There are probably only two tag sets that are the most widely used:
  - Penn Tag Set
  - EAGLES Tag Set

Part-of-Speech Tagging (4)
PENN TAG SET

- In the case of American English, the Penn tag set, developed in the Penn Treebank project at the University of Pennsylvania is probably the most common choice. It is also frequently preferred by automatic tagging systems, since it is largely similar to the Brown Corpus tag set, but much smaller.

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

Part-of-Speech Tagging (5)

EAGLES TAG SET

- On the other hand, in Europe, tag sets from the EAGLES (Expert Advisory Group on Language Engineering Standards) Guidelines have wide use and include versions for multiple languages.

ADJETIVOS			
Pos.	**Atributo**	**Valor**	**Código**
1	Categoría	Adjetivo	A
2	Tipo	Calificativo	Q
		Ordinal	O
		-	0
3	Grado	-	0
		Aumentativo	A
		Diminutivo	C
		Superlativo	S
4	Género	Masculino	M
		Femenino	F
		Común	C
5	Número	Singular	S
		Plural	P
		Invariable	N
6	Función	-	0
		Participio	P

VERBOS			
Pos.	**Atributo**	**Valor**	**Código**
1	Categoría	Verbo	V
2	Tipo	Principal	M
		Auxiliar	A
		Semiauxiliar	S
3	Modo	Indicativo	I
		Subjuntivo	S
		Imperativo	M
		Infinitivo	N
		Gerundio	G
		Participio	P
4	Tiempo	Presente	P
		Imperfecto	I
		Futuro	F
		Pasado	S
		Condicional	C
		-	0
5	Persona	Primera	1
		Segunda	2
		Tercera	3
6	Número	Singular	S
		Plural	P
7	Género	Masculino	M
		Femenino	F

Syntactic Chunking

- Intuitions to define a chunk:
  - The strongest stresses in a sentence fall one to a chunk,
  - Pauses are most likely to fall between chunks.
- Chunks can be understood as textual units of adjacent word tokens which can be mutually linked through unambiguously identified dependency chains with no recourse to idiosyncratic lexical information. Chunks present a set of properties:

Chunks are non-overlapping regions of text.

(Usually) each chunk contains a head, with the possible addition of some preceding function words and modifiers

Chunks are non-recursive, a chunk cannot contain another chunk of the same category.

Chunks are non-exhaustive, some words in a sentence may not be grouped into a chunk.

Noun groups and verb groups are chunks.

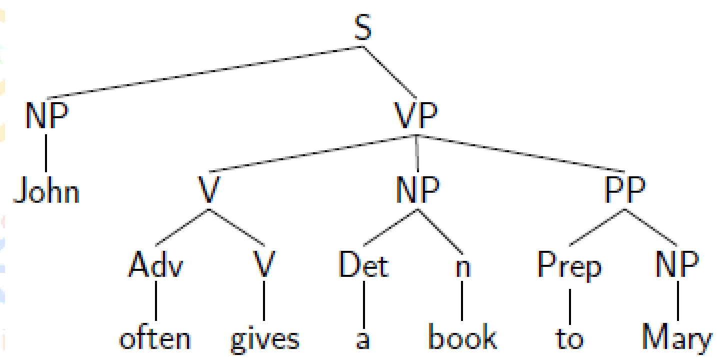
Syntactic Chunking (2)

- Categories can be identified when chunking. Most common categories include: **Noun Phrases, Verb Phrases, Prepositional Phrases, Adjectival Phrases and Adverbial chunks**
- Different notations have been developed so far. One of the most common is the notation used in the Conference on Computational Natural Language Learning in 2000 (or CoNLL-2000).
- The chunk tags contain the name of the chunk type and the special mark B-CHUNK is used for the first word of the chunk, while I-CHUNK is used for each other word in the chunk.

He	PRP	B-NP
reckons	VBZ	B-VP
the	DT	B-NP
current	JJ	I-NP
account	NN	I-NP
deficit	NN	I-NP
will	MD	B-VP
narrow	VB	I-VP
to	TO	B-PP
only	RB	B-NP
#	#	I-NP
1.8	CD	I-NP
billion	CD	I-NP
in	IN	B-PP
September	NNP	B-NP
.	.	O

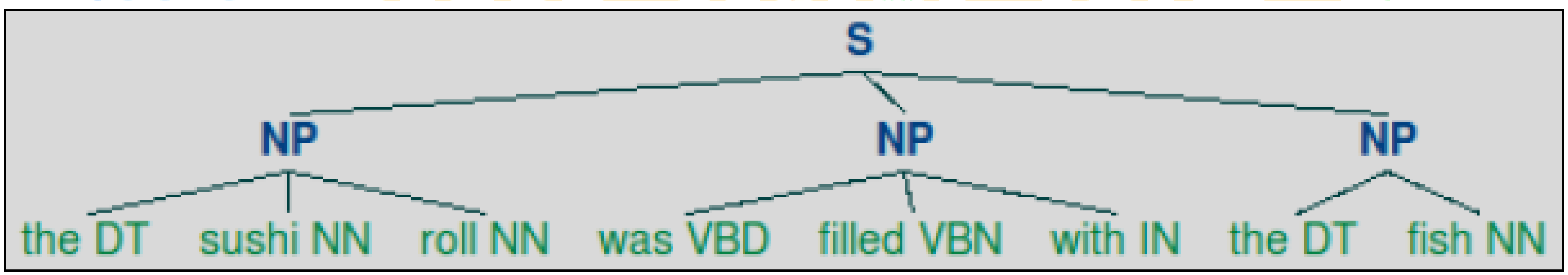
Chunking v/s Parsing

- Chunking is the process of dividing a text in syntactically correlated parts of words. From this it follows that chunking is an intermediate step towards full parsing.
- A parser is capable of assigning a syntactic structure (i.e. discovering the structural relationships between words and phrases) to a string on the basis of a grammar, used to describe the syntax of a language.



Chunking v/s Parsing (2)

- In contrast to parsing, chunking:
  - Yields flatter structures than full parsing, generally using fixed tree depth (max depth of 2 vs. arbitrarily deep trees)
  - Does not try to deal with all of language nor attempt to resolve all semantically significant decisions.



Section 3.5

Web Page Content Classification

Text Classification

- Assign a label to a text. Ex:
  - Classify as a Political Document
  - Classify as a particular Product Marketing page
  - Classify as a Study on Molecular Biology
  - Etc…
- The label could be:
  - Pre-defined: By the direction of the study -> SUPERVISED LEARNING
  - Unknown: To discover! -> UNSUPERVISED LEARNING

Practical Uses

- Extracting Domain Specific Information
  - Grouping documents in different domains.
  - Finding the most representative
- Learn reading interests of users
- Automatically classification of e-mail
- On-line New Event Detection:
  - Opinion blog scanners.
  - Social activities detection.

Text Classification

- A text classifier:
  - Given a document *d*, return a scalar value with a category [Sebastiani99].

$$CSV_i : D \rightarrow [0,1] \quad c_i \in C / \bigcup c_i = C$$

- The function is known as "Categorization Status Value", $CSV_i(d)$.
- The algorithm takes different expressions, according with the classifier in use.
- For instance, it can be a probability approach [Lewis92] basis on Naive Bayes theorem or a distance between vectors in a r-dimensional space [Schutze95].

Text Classification (2)

- Text classifiers have been implemented with semi-automatic or full-automatic [Asirvatham05] approaches, like:
  - K-nearest neighbor [Kwon03]
  - Bayesian models [McCallum98]
  - Support Vector Machines [Joachims97]
  - Artificial Neural Networks [Honkela97]
  - Decision Trees [Apte04] .
  - MAXENT algorithm [McCallum99]

Text Classification (3)

- The web pages classification algorithms can be grouped in [Asirvatham05] :
  - Manual categorization -> too expensive
  - Applying clustering approaches. Previous to classify the web pages, a clustering algorithm is used to find the possible clusters in a training set.
  - Meta tags: It use the information contained in the web page tags (<META name=``keywords''> and <META name=``description'' >).
  - Text content based categorization.
  - Link and content analysis: It is based on the fact that the hyperlink contain the information about which kind of pages is pointed (href tag)

Issues on text classification

- High dimensionality.
- Ambiguity and text sensitivity: “the main course was explained in the thai food course”
- Need a large labeled example for training the models.
- Unstructured data

Further refining for text mining

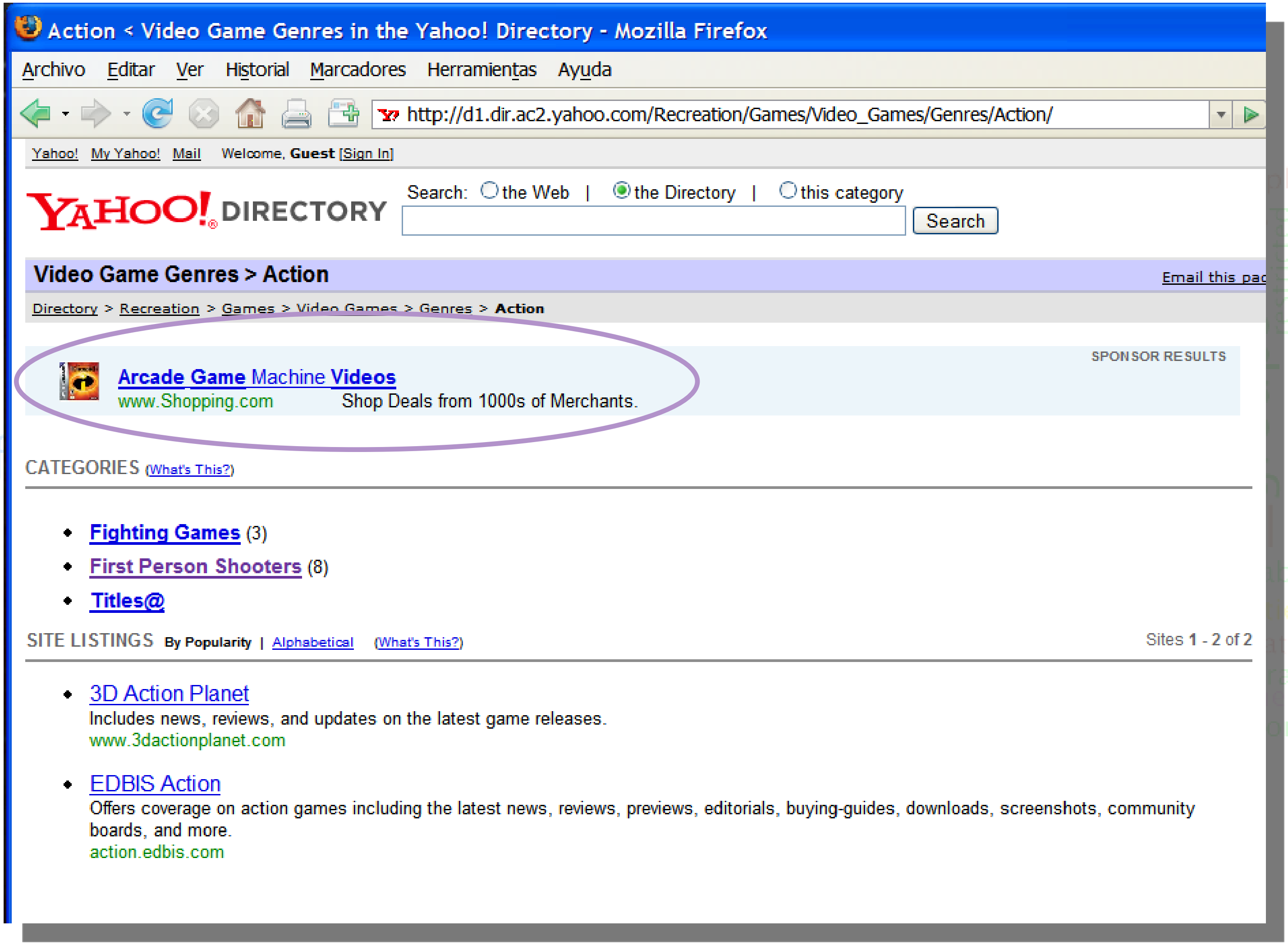
- Semantic Processing
  - Extracting meaning
  - Named entities extraction (people names, company names, locations, …)
  - Phrase recognition
  - Tagged phrases
- The semantic process result in more complex numeric vector structure with nested relationship (tree-like).

We can use Natural Language Processing tools!

Hierarchies: Natural Human Classification

- ¿A number is something useful as classification? -> No
- Human need always to have a “context” for classification of something.
- This “context” contains classes, but the content also need to have a “context”.
- Create a tree-like or directory-like hierarchy of contexts.
- In the Web, these hierarchies are called “Directories”

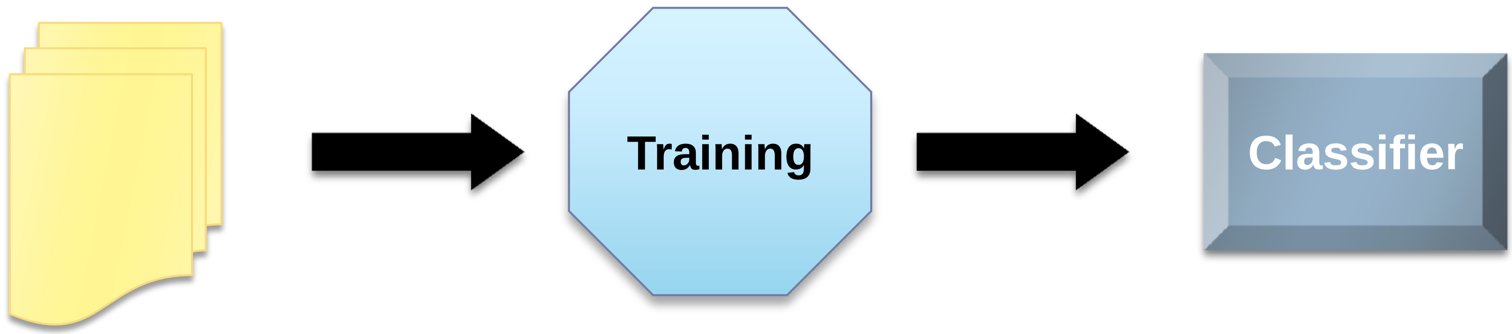
Hierarchies: Yahoo Directory



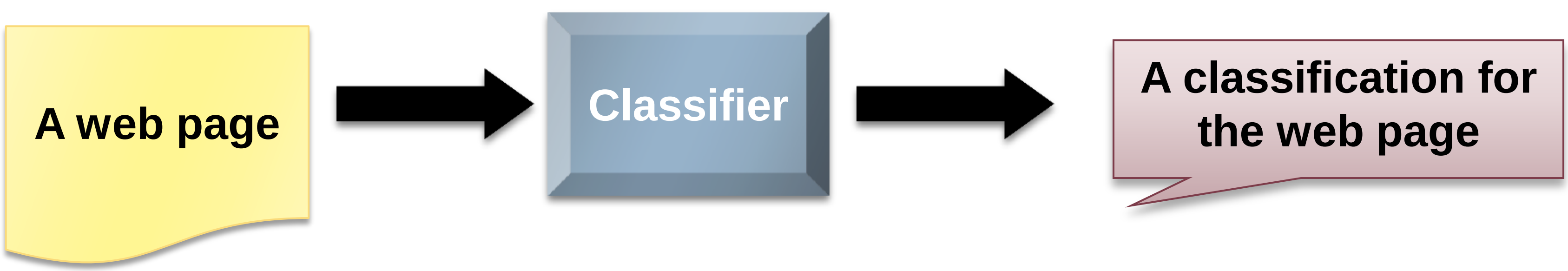
Web Hierarchy are Web Directories

- **Example:**
  - **Yahoo Directories, Google directories.**
  - Each time that we perform a search the result appears as belonging to a directory structure.
- An open source web directory information is available on the DMOZ project (from Netscape) http://www.dmoz.org/
- You can download free 1 Gb of human web classification on RDF format.
- Some software allows to parse this format and translate to database format. The project dmoz2mysql http://sourceforge.net/projects/dmoz2mysql/

Web directory importance: Building a Training Set (Supervised)



**Labeled set of web pages
from web directory classification**



Section 3.6

Algorithms for text classification: A revision of the current literature

K-means clustering

- "Text categorization based on k-nearest neighbor approach for Web site classification", O. Kwon, J. Lee, 2003.
- Given:
  - Set of word vector (TFIDF value)
  - Similarity measure (cosine)
  - An estimation of the number of classes
- For each class initialize randomly the Centroid.
- Iterate assigning the nearest group for each page and recalculate the Centroid.
- Finish when Centroid converge.

Support Vector Machines for Text Classification

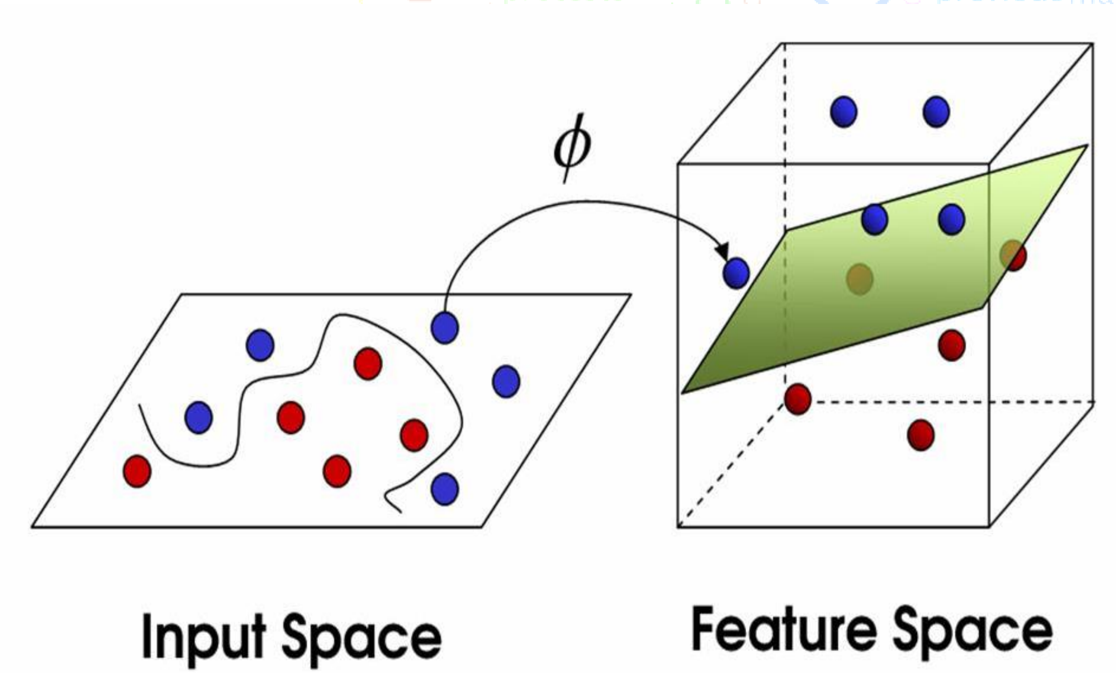
- T. Joachims, "*Text Categorization with Support Vector Machines: Learning with Many Relevant Features*". 1997.
- Word Vector are sometimes very high dimensional, sometimes 10000 different keyword per document.
- **Feature selection**: the process that allows to choose the correct keyword (in this case) in order to have a low dimensional model.
- Chi-square, Information Gain, are commonly used.
- Word vector are also sparse.
- … but we could lose valuable information for clustering

Support Vector Machines for Text Classification (2)

- SVM in short:
  - As a result define “hyperplanes” that separate the data in the different classes.
  - The “hyperplane” are defined to maximize the distance to the training point.
  - The methodology generalize to non-linear geometry, where the dot product between vector are nonlinear kernel function.
  - The resulting set of hyper-plane are not plane, there are curved hyper-surfaces

Support Vector Machines for Text Classification (2)

- SVM in short:



SVM for Text Classification

- SVM: Can handle high dimensional space.
- Comparing between method using microaveraging [Joachims, 1998] threshold performance measure:

Other classifiers | Polynomial Kernel | Gaussian Kernel

class	Bayes	Rocchio	C4.5	k-NN	SVM (poly) $d =$ 1	2	3	4	5	SVM (rbf) $\gamma =$ 0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined **86.0**					combined **86.4**			

SVM has the best score

SVM for Text Classification

- Between many others

Other classifiers | Polynomial Kernel | Gaussian Kernel

class	Bayes	Rocchio	C4.5	k-NN	SVM (poly) $d =$ 1	2	3	4	5	SVM (rbf) $\gamma =$ 0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

SVM has the best score

Section 3.7

Clustering for groups having web page text content

Clustering: Unsupervised method

- Clustering is a process of finding natural groups in a unsupervised way.
- To group web pages allows perform efficient searching task and semi-automatic or full-automatic document's categorizations.
- The clustering techniques need a similarity measure in order to compare two vectors by common characteristics [Strehl00].

Clustering

- It is necessary a similarity of distortion measure to compare the vectors in a training set.
- For instance a simple distance like the angle's cosine between two pages in a vector representation.

$$dp(wp_i, wp_j) = \cos\theta = \frac{\sum_{k=1}^{R} wp_{ki}\, wp_{kj}}{\sqrt{\sum_{k=1}^{R} (wp_{ki})^2}\,\sqrt{\sum_{k=1}^{R} (wp_{kj})^2}}$$

Clustering (2)

- For document clustering, more complex and semantic based similarity have been proposed [Strehl00].
- Let $C = \{c_1, \ldots, c_l\}$ be the set of clusters extracted from WP.
- Since the hard clustering point of view

$$\exists! c_k \in C \,/\, wp^i \in c_k$$

(it belongs to a only one class)

Clustering (3)

- Whereas in soft clustering, a vector can belong to two or more clusters [Karypis99, Koutri04] .
- Several document clustering algorithms have appeared in the last years [Feldman95, Willet88] .
- An interesting approaches is the utilization of K-means and its variations in overlapping clusters, known as Fuzzy C-means [Jang97] .
- In these cases a word vector could belong to several classes.

Effect of different similarity measures on clustering

- Strehl, Gosh, Mooney, “Impact of Similarity Measures on Web-page Clustering”, AAAI-2000.
- We have already known the cosine measure. But there are others.

$$s^{(C)}(x_a, x_b) = \frac{x_a^{T} x_b}{\|x_a\|_2^2 \|x_b\|_2^2}$$

Effect of different similarity measures on clustering (2)

- Pearson Measure: where $\overline{x}$ denotes the average.

$$s^{(P)}(x_a, x_b) = \frac{1}{2}\left(\frac{(x_a - \overline{x}_a)^{\mathrm{T}}(x_b - \overline{x}_b)}{\|x_a - \overline{x}_a\|_2 \|x_b - \overline{x}_b\|_2} + 1\right)$$

- Jaccard Measure:

$$s^{(J)}(x_a, x_b) = \frac{x_a^{\mathrm{T}} x_b}{\|x_a\|_2^2 \|x_b\|_2^2 - x_a^{\mathrm{T}} x_b}$$

- Euclidean Measure:

$$s^{(E)}(x_a, x_b) = \|x_a - x_b\|_2$$

Effect of different similarity measures on clustering (3)

- Observation:
  - Euclidean measure have the worst results, even bad than a random clustering.
  - Cosine and Jaccard measure are the best ones.
- The Jaccard measure appears as an alternative.
- Its represents an approximation of the quotient information of (A and B) versus (A or B).

A relaxed problem: few labeled document.

- If we have a very few labeled document for training, the problem is close to unsupervised clustering.
- Nigam, McCallum, Thrun, Mitchel, “Text Classification from Labeled and Unlabeled Documents using EM”, Machine Learning, 2000.
- Using Naïve Bayes Classifiers they infer the label of the others iteratively until the classifiers converge.

Section 3.8

Web Opinion Mining

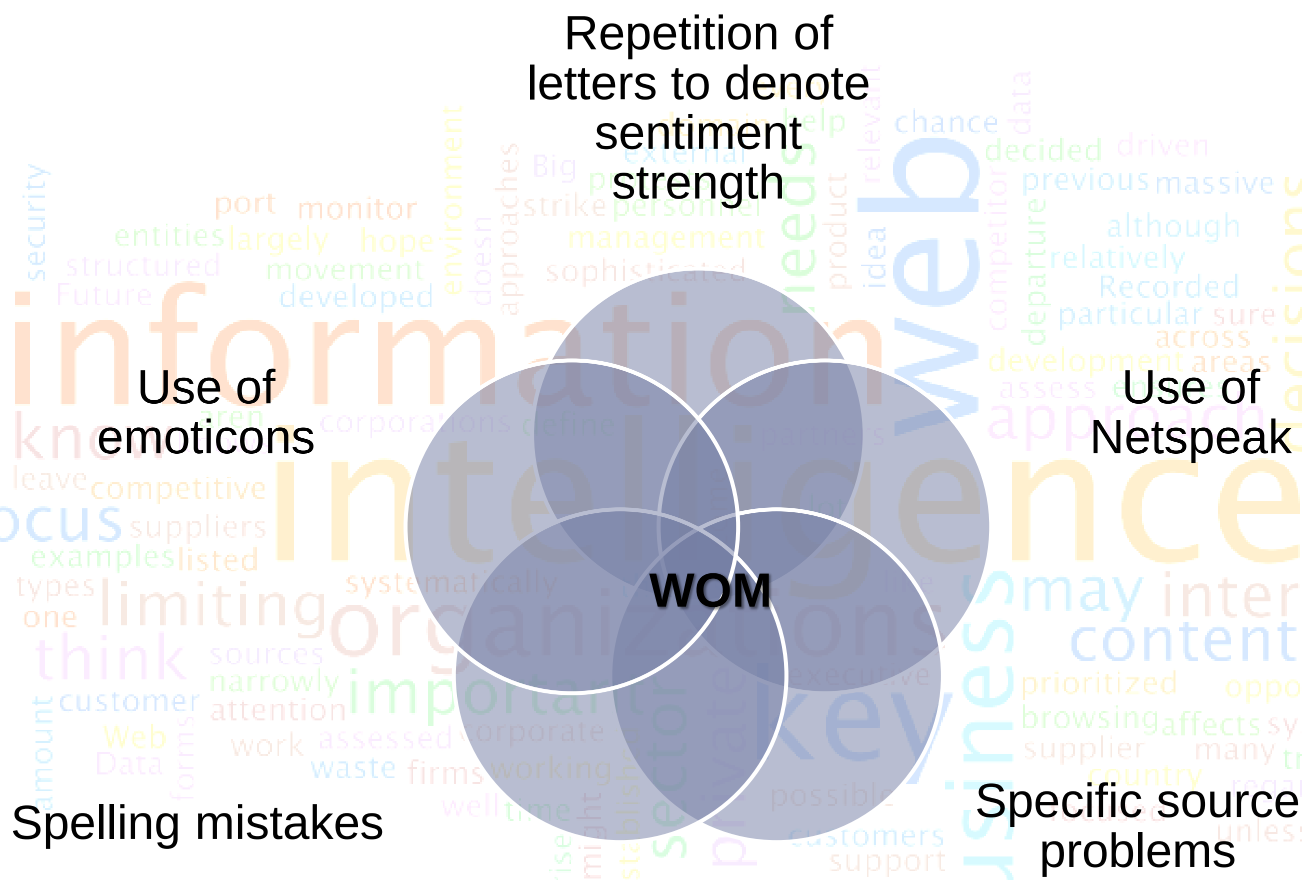
Introducing Opinion Mining

- The computational study of opinions, sentiments and emotions expressed in text (Liu 2010).
- It was born as a discipline mostly due to the development of the Web 2.0. Because of the explosive growth of social media, people now use these tools to make better decisions (Park & Kim, 2009) (Shin, Hanssens, Kim, & Gajula, 2011) (Zhou & Chaovalit, 2007)
- Opinions are important because they are key influencers of our behaviors, our beliefs and perceptions of reality, and the choices we make, are to a considerable degree conditioned on how others see and evaluate the world.

Toward a unique WOM definition

- WOM is a new tool and has a long way to walk.
- Giving a unique definition for WOM is not a simple task because the **process final objective is still unclear**.
- There are many ways to embrace this problem in literature.
  - Aspect-Based
  - Document Level
  - Sentence Level
  - Etc.

Generic Challenges



Let's check some of these…

Aspect-based Opinion Mining (1)

- Proposes the extraction of product "features" in opinions **(Liu 11).**
- Opinions are modeled as 5-tuples:

$$(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$$

- e_i = Entity
- a_{ij} = Aspect
- oo_{ijkl} = Sentiment Orientation
- h_k = Opinion holder's name
- t_l = Time/Date

Aspect-based Opinion Mining (2)

- Problem: Given a set of documents *D,* find all the opinion tuples in D.
- Methodology proposes

Step1: Entity Extraction and agrupation.

Step2: Aspect Extraction and agrupation.

Step 3: Extraction of opinion holder and time/date.

Step 4: Sentiment Orientation determination.

Step 5 : Generation of all the opinion 5-tuples.

Non Aspect-based Opinion Mining

- Includes all the other kinds of opinion mining which do not divide the text into subtopics.
- In general, they simply consider the text as a big object or increase granularity analyzing each paragraph, sentence or phrase.
- It is possible so consider a generic three-phase process, which is introduced by (Plantiè et al. 09)
  - **Phase 1: Corpora Acquisition Learning Phase**
    - Automatically extract documents containing positive and negative opinions from the Web, for a specific domain.
  - **Phase 2: Adjective Extraction Phase**
    - Automatically extract sets of relevant positive and negative adjectives.
  - **Phase 3: Classification**
    - Classify new documents using the sets of adjetives obtained in the previous phase.

Step 1: Aspect identification/extraction

- Only for aspect-based approaches. The concept of *aspect* comes from the idea that, in general, opinions can be expressed about anything: a product, service, organization, etc.
- The set of aspects underpinning the text could or could not be known previously, which implies that different problems need to be solved.
- Also, different people could refer to the same aspect with different words, which brings additional problems to the task.

Step 1: Aspect identification/extraction (2)

- **Pure NLP Techniques:** Some approaches attempt to identify features in the opinion text with the help of NLP-based techniques. Part-of-speech (POS) tagging and syntax tree parsing are very common starting points for aspect discovery. In most of the cases, annotated opinion texts are then analyzed using classic data mining techniques. Examples of this are the works of Lu 2009, Popescu and Etzioni 2005 and Hu and Liu 2004.
- **Mining and Statistical Techniques**: Classic data mining approach on finding aspects are also used, usually as an attempt to compensate the weaknesses of a pure NLP-based technique. This approach shows reasonable performance, especially with product reviews. Examples: Archak et al. 2007 and Decker and Trusov 2010.
- **Ontology-Supported Techniques:** Some authors look for aspects by exploiting ontologies, a representation of knowledge as a set of concepts within a domain, and the relationships between pairs of concepts.
  - The set of possible aspects is given and the problem of extracting them transforms into matching.
  - Extracted features correspond exclusively to terms contained in the ontology.

Step 2: Subjectivity Classification

- Aims at different sub-segments of the text, trying to differentiate sub-segments that include any opinion or evaluation from the ones that do not.
- The process can be applied to documents, paragraphs or sentences.
- Subjectivity classification is different from sentiment classification (next slide) in that the former only aims at finding if an opinion is present or not and does not attempt to identify the orientation of these opinions

Step 2: Subjectivity Classification (2)

- Existing literature is vast:
  - Hatzivassiloglou and Wiebe 2000 : word clustering
  - Riloff and Wiebe 2003
  - Yu and Hatzivassiloglou 2003: Bayesian approach
  - Etc.
- The importance of subjectivity classification is that it could be used as an input data preprocessing step for sentiment classification.
- By filtering out objective sentences in advance of sentiment classification, subjectivity classification can increase the accuracy of sentiment classification.

Step 3: Sentiment Classification

- The process that aims to determine the sentiment orientation of a document, or part of a document.
- The objective of this phase is to classify each document or document segment into two different categories, *Positive* or *Negative*.
- Techniques can be classified into three main groups:

Classification Based on Machine Learning: In general, any supervised ML method can be applied for this task, the most used ones being Naïve Bayes and Support Vector Machines.

Rule Based: Instead of using a standard machine learning method, researchers have also proposed several custom techniques specifically for sentiment classification, like score functions and aggregation methods.

Step 3: Sentiment Classification (2)

- It could also include determining the degree of sentiment in an arbitrary scale.
- Most current techniques that involve Machine Learning algorithms and are based on features like:

- Terms and their frequency
- Part of Speech Tagging (POS)
- Opinion words and phrases
- Opinion Rules
- Negations
- Syntactic dependency

Step 4: Summarization and Visualization

- To analyze a high number of opinions, automatic systems are required. For humans it's impossible to read and process thousand of opinions to make a decision.
- **Document Based Summarization** **(Chen, Ku, Liang 04)**
  - Count possitive-negative documents
  - Generate automatic short text reviews

Positive	Ahmad Rejai Al-Jundi, Assistant Secretary General of the Islamic Organization, declared earlier that the seminar would be aimed at shedding light on medical and legal aspects of the internationally controversial issue and seeking to take a stand on it.
Negative	Dolly the cloned sheep is only 3, but her genes are already showing signs of wear and she may be susceptible to premature aging and disease -- all because she was copied from a 6-year-old animal, Scottish researchers say.

Table 19. Detailed Opinion Summary of TREC

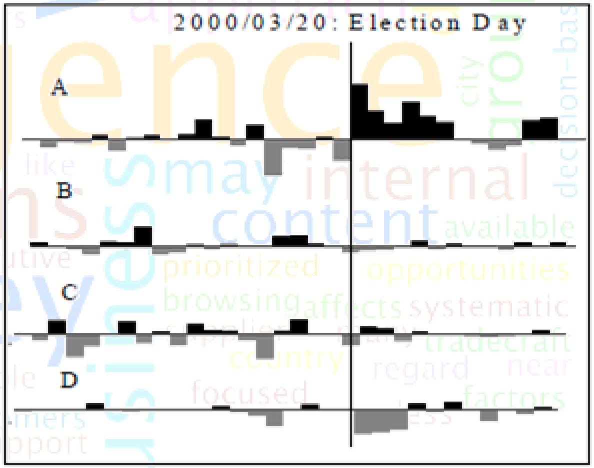
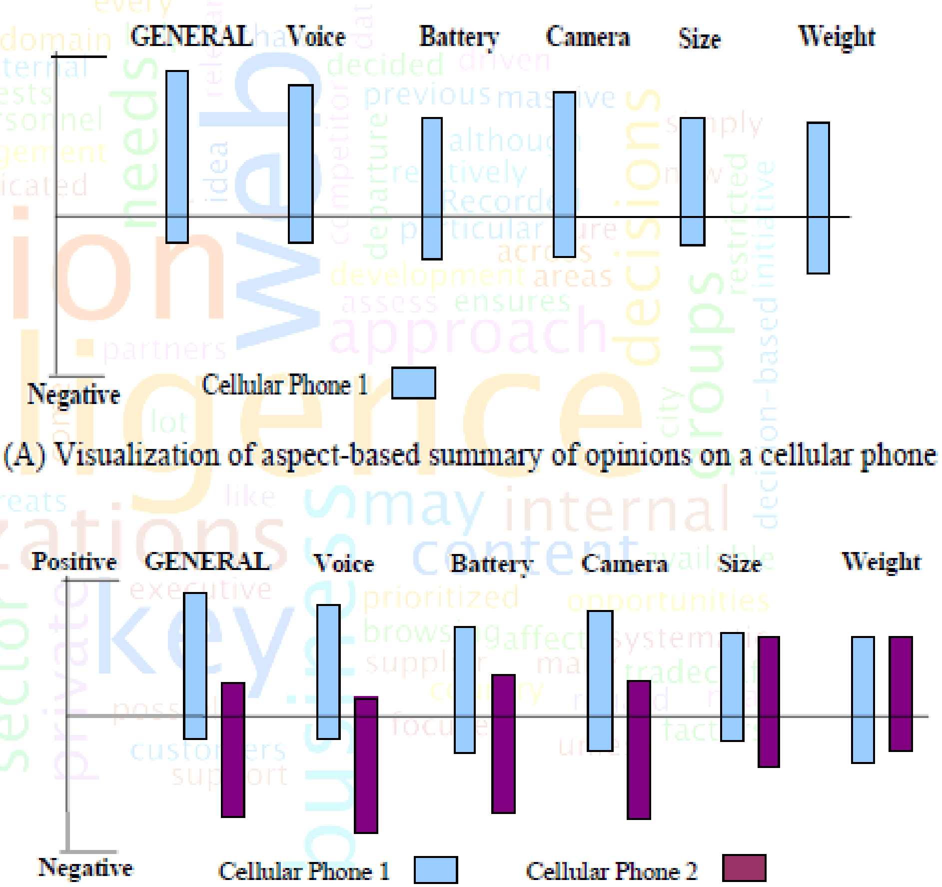


Figure 1. Opinions towards four persons

)

Step 4: Summarization and Visualization (2)

- **Aspect-Based Summarization** **(Liu 11)**
  - Count possitive and negative opinions about a specific prodcut and its aspects.
  - Plot results in a bar graph, where y-axis measures the number of possitive or negative opinions for each aspect.
  - This technique enables the possibility of comparing similar products.



(A) Visualization of aspect-based summary of opinions on a cellular phone

(B) Visual opinion comparison of two cellular phones

Section 3.9

Applications

Applications

WEBSOM

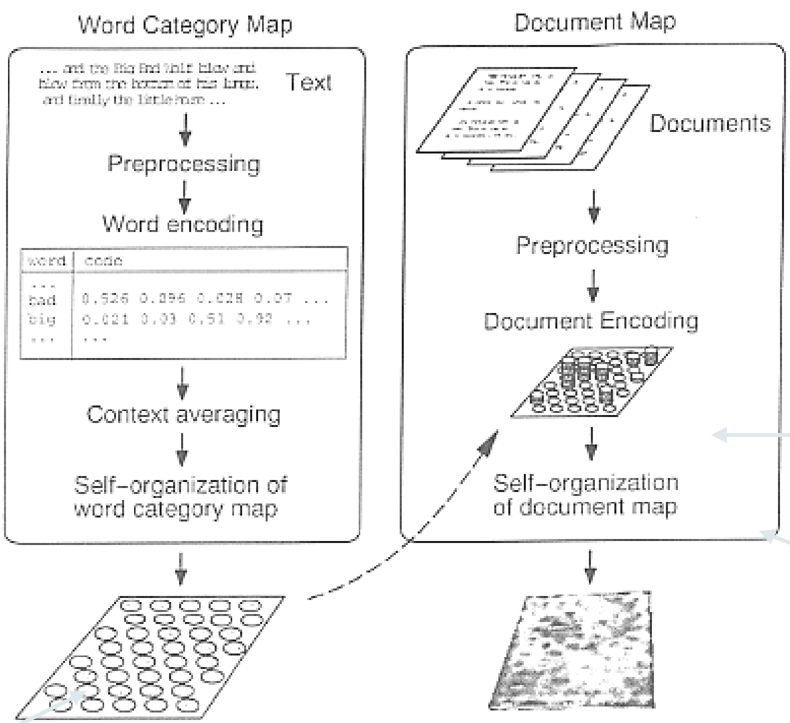
Automatic web page text summarization

Extraction of key-text component from web pages

WEBSOM

- It is a means for organizing miscellaneous text documents into meaningful maps for exploration and search.
- It is based on SOM (Self-Organizing Map) that automatically organizes documents into a two-dimensional grid so that related documents appear close to each other
- http://www.cis.hut.fi/websom

WEBSOM (2)



All words of document are mapped into the word category map

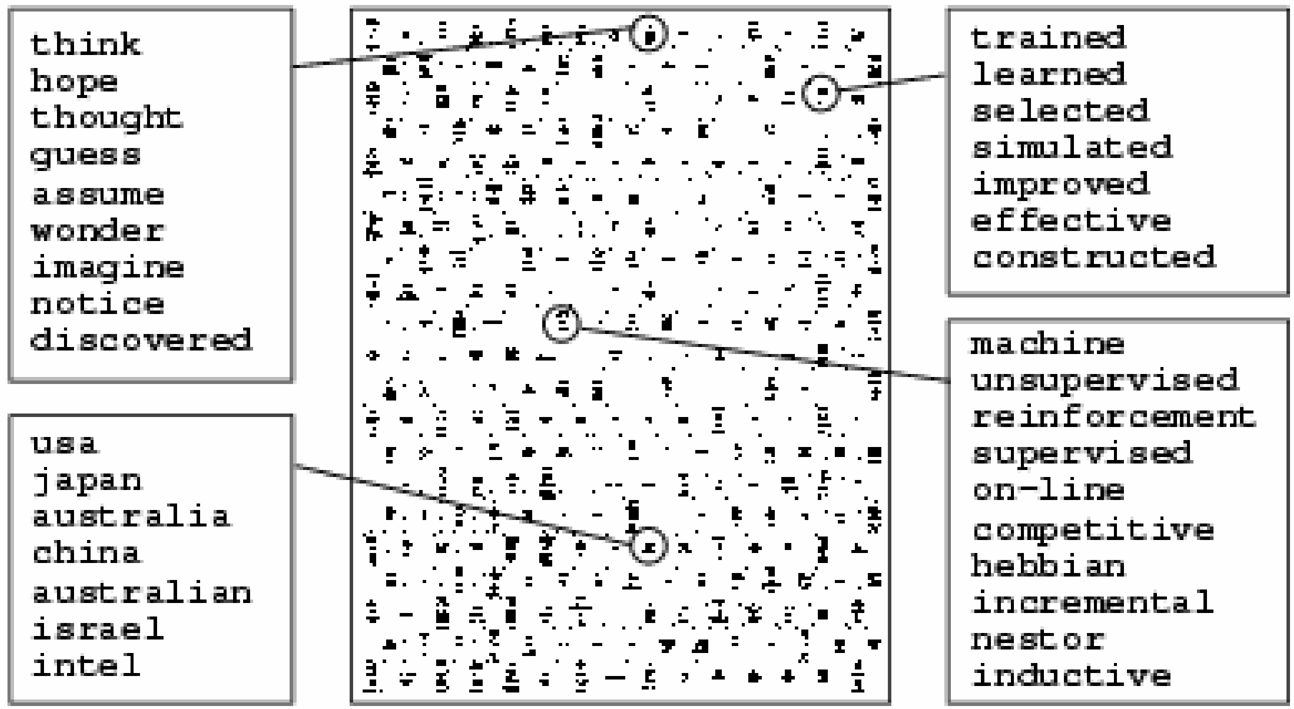
Histogram of "hits" on it is formed

Self-organizing map:
Largest experiments have used:
- Word-category map
  315 neurons with 270 inputs each
- Document-map
  104040 neurons with 315 inputs each

Self-organizing semantic map: 15x21 neurons
Interrelated words that have similar contexts appear close to each other on the map

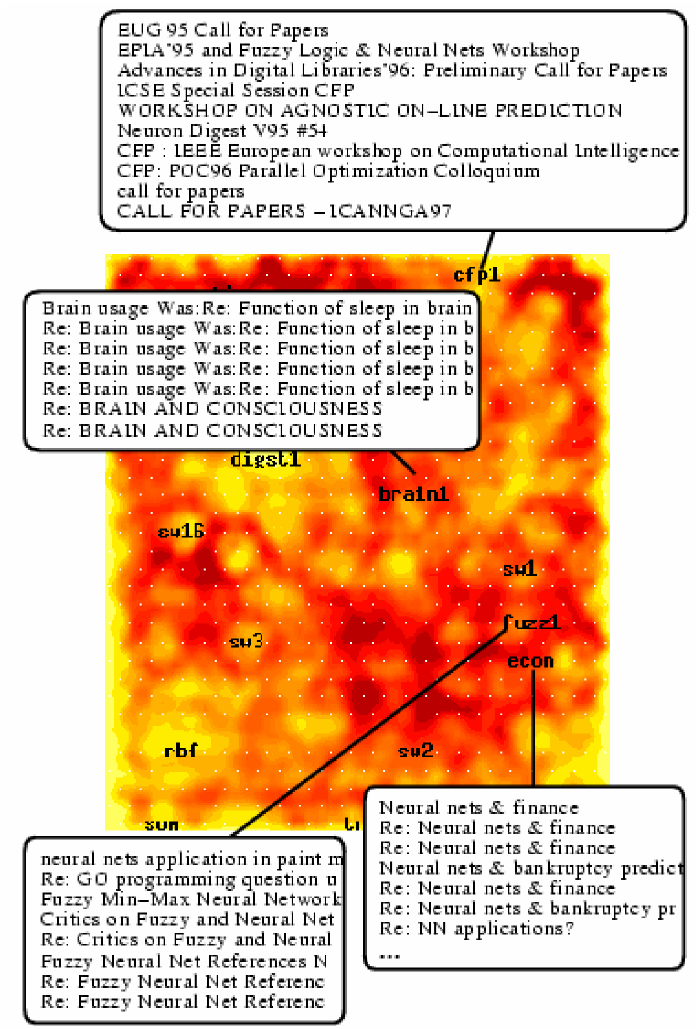
Training done with 1124134 documents

Word categories



Each inset shows the words that have been mapped into one word category

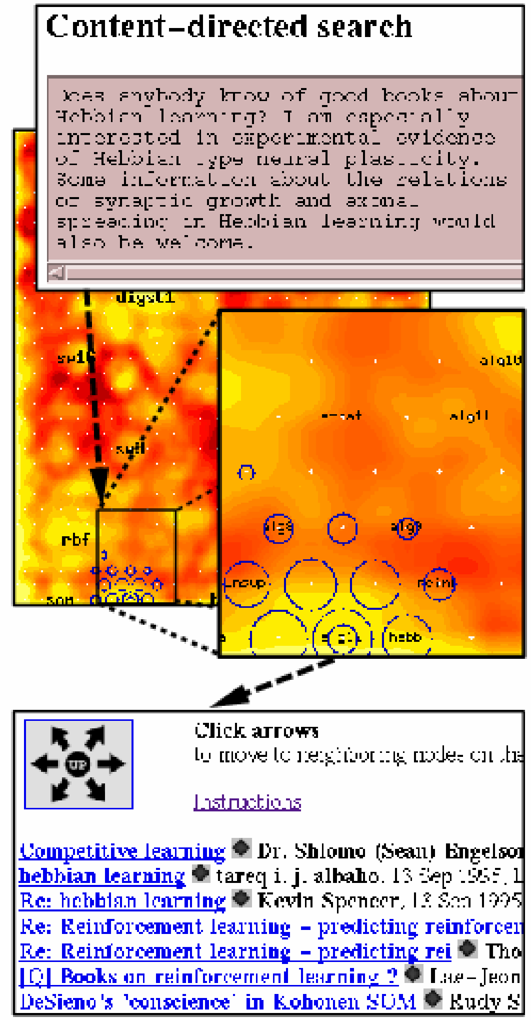
A map of documents



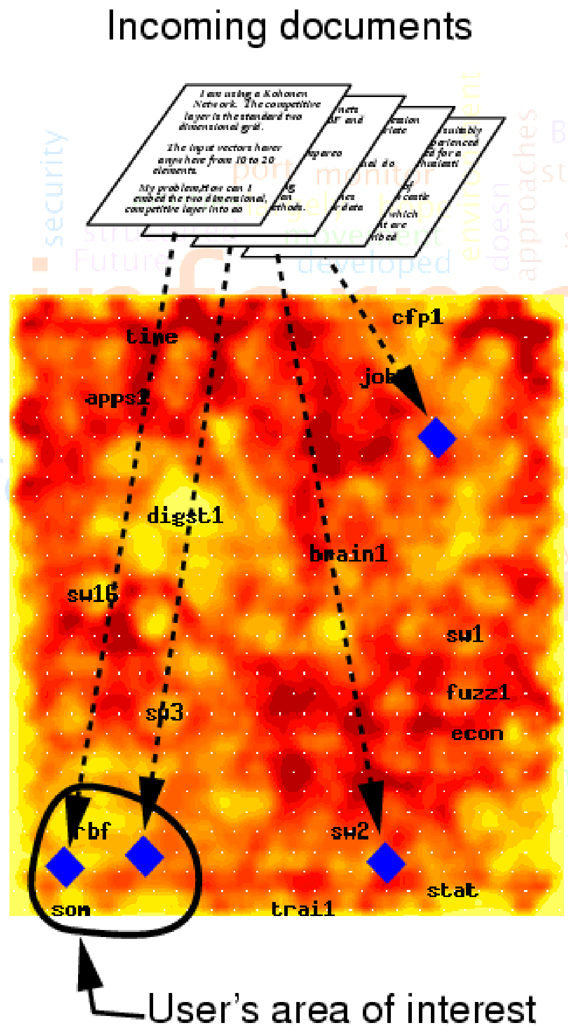
- A set of documents related with neural networks is mapped by using WEBSOM method.
- Browsing for the interface, it is possible to see the "labels" for documents

Sample search

- A new document or any document's description can be used for finding related documents.
- The circle on the map denote the location of the most representative document for the question.



How to use the Maps



- As a filter that notifies the user of interesting documents
- As a searching engine
- A new index method

Automatic web page text summarization

- The goal is to construct automatically summaries of a natural-language document [Hahn00] .
- In many case the web pages only contain few words and the page could contain non-textual elements (e.g. video, pictures, audio, etc.) [Amitay00] .
- In text summarization research, there are three major approaches [Mani99] :
  - Paragraph based
  - Sentence based
  - Using natural language cues in the text.

Types of summaries

- Purpose
  - Indicative, informative, and critical summaries
- Form
  - Extracts (representative paragraphs/sentences/phrases)
  - Abstracts: “a concise summary of the central subject matter of a document” [Paice90].
- Dimensions
  - Single-document vs.. multi-document
- Context
  - Query-specific vs.. query-independent

Genres

- Headlines
- Outlines
- Minutes
- Biographies
- Abridgments
- Sound bites
- Movie summaries
- Chronologies,
- Etc.

[Mani and Maybury 1999]

What does summarization involve?

- Three stages (typically)
  - Content identification
  - Conceptual organization
  - Realization

Kupiec et al. 95

- Uses Bayesian classifier:

$$P(s \in S \mid F_1, F_2, \ldots F_k) = \frac{P(F_1, F_2, \ldots F_k \mid s \in S)\, P(s \in S)}{P(F_1, F_2, \ldots F_k)}$$

Assuming statistical independence:

$$P(s \in S \mid F_1, F_2, \ldots F_k) = \frac{\prod_{j=1}^{k} P(F_j \mid s \in S)\, P(s \in S)}{\prod_{j=1}^{k} P(F_j)}$$

Kupiec et al. 95

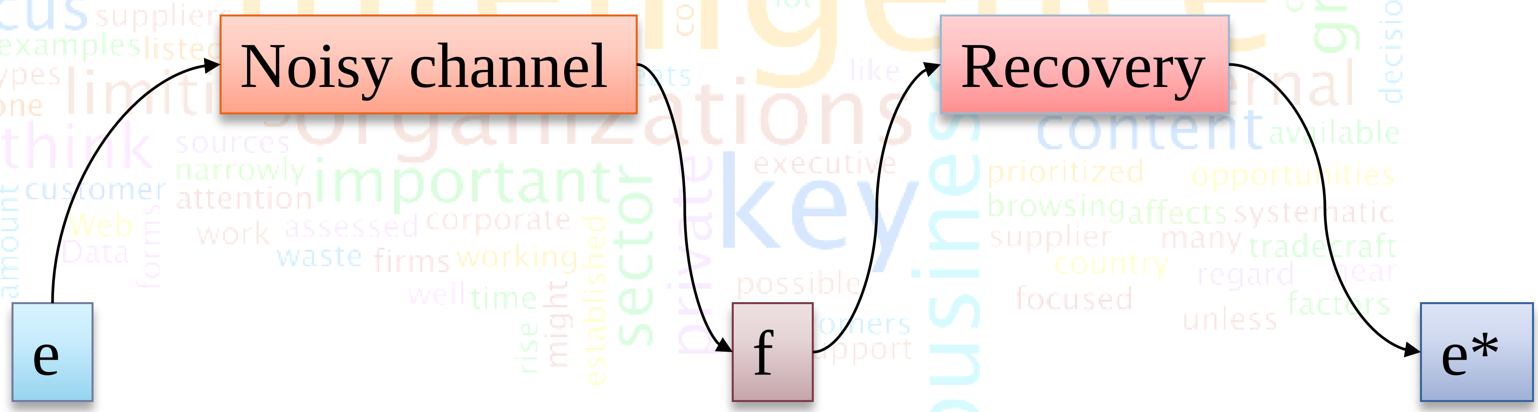
- Performance:
  - For 25% summaries, 84% precision
  - For smaller summaries, 74% improvement over Lead

Osborne 02

- Maxent (loglinear) model – no independence assumptions
- Features: word pairs, sentence length, sentence position, discourse features (e.g., whether sentence follows the “Introduction”, etc.)
- Maxent outperforms Naïve Bayes

Language modeling

- Source/target language
- Coding process



Language modeling

- Source/target language
- Coding process

$$e^* = \underset{e}{argmax}\ p(e|f) = \underset{e}{argmax}\ p(e)\ .\ p(f|e)$$

$$p(E) = p(e_1).p(e_2|e_1).p(e_3|e_1e_2)\ldots p(e_n|e_1\ldots e_{n-1})$$

$$p(E) = p(e_1).p(e_2|e_1).p(e_3|e_2)\ldots p(e_n|e_{n-1})$$

Summarization using LM

Source language

- Full document

Target language

- Summary

Berger & Mittal 00

- Gisting (OCELOT)

$$\boxed{g^* = \underset{g}{\operatorname{argmax}}\ p(g|d) = \underset{g}{\operatorname{argmax}}\ p(g)\ .\ p(d|g)}$$

- Content selection (preserve frequencies)
- Word ordering (single words, consecutive positions)
- Search: readability & fidelity

Berger & Mittal 00

- Limit on top 65K words
- word relatedness = alignment
- Training on 100K summary+document pairs
- Testing on 1046 pairs
- Evaluation: word overlap (0.2-0.4)
- Transilingual gisting is possible
- No word ordering

Berger & Mittal 00

Sample output:

Audubon society atlanta area savannah georgia chatham and local birding savannah keepers chapter of the audubon georgia and leasing

Extraction of key-text components from web pages

- The key-text components are parts of an entire document
- **Key-text:** *A paragraph, phrase and inclusive a word*, that contain **significant information** about a particular topic, from the *web site user point of view*.
- A **web site keyword** is *"a word or possibly a set of words that make a web page more attractive for an eventual user during his visit to the web site""* [Velasquez05b].

Extraction of key-text components from web pages

- The assumption is that there exists a ***correlation*** *between the* ***time that the user spent in a page*** *and his/her* ***interest*** *in its content* [Velasquez04b] .
- Usually, the **keywords** in a web site have been related with the **"most frequently used words".**
- In [Buyukkokten01] a method to extract keywords from a huge set of web pages is introduced.

Summary

- The **vector space model** is a recurrent *method to represent a document as a feature vector*.
- Because the set of words used in the construction of the web site could be too big, it is necessary to apply a **stop word cleaning** and **stemming process**.
- A web page content is different from a common document. In fact, a web page contains **semi-structured text**, i.e., tags that give additional information about the text component.
- Also a page could contain *pictures, sounds, movies, etc.*
- Sometimes, the page text content is a **short text** or even a **set of unconnected words**.