Preparation to Watson Tutorial

Alessandro Moschitti
DISI
University of Trento
Email: moschitti@disi.unitn.it

Outline

- Motivation
  - Question Answering vs. Search Engines
- Information Retrieval Techniques
  - Search Engines
  - Vector Space Model
  - Feature Vectors and Feature Selection
  - Text Categorization
  - Measures
- Machine Learning Methods
  - Classification
  - Ranking
  - Regression (logistic)

Outline

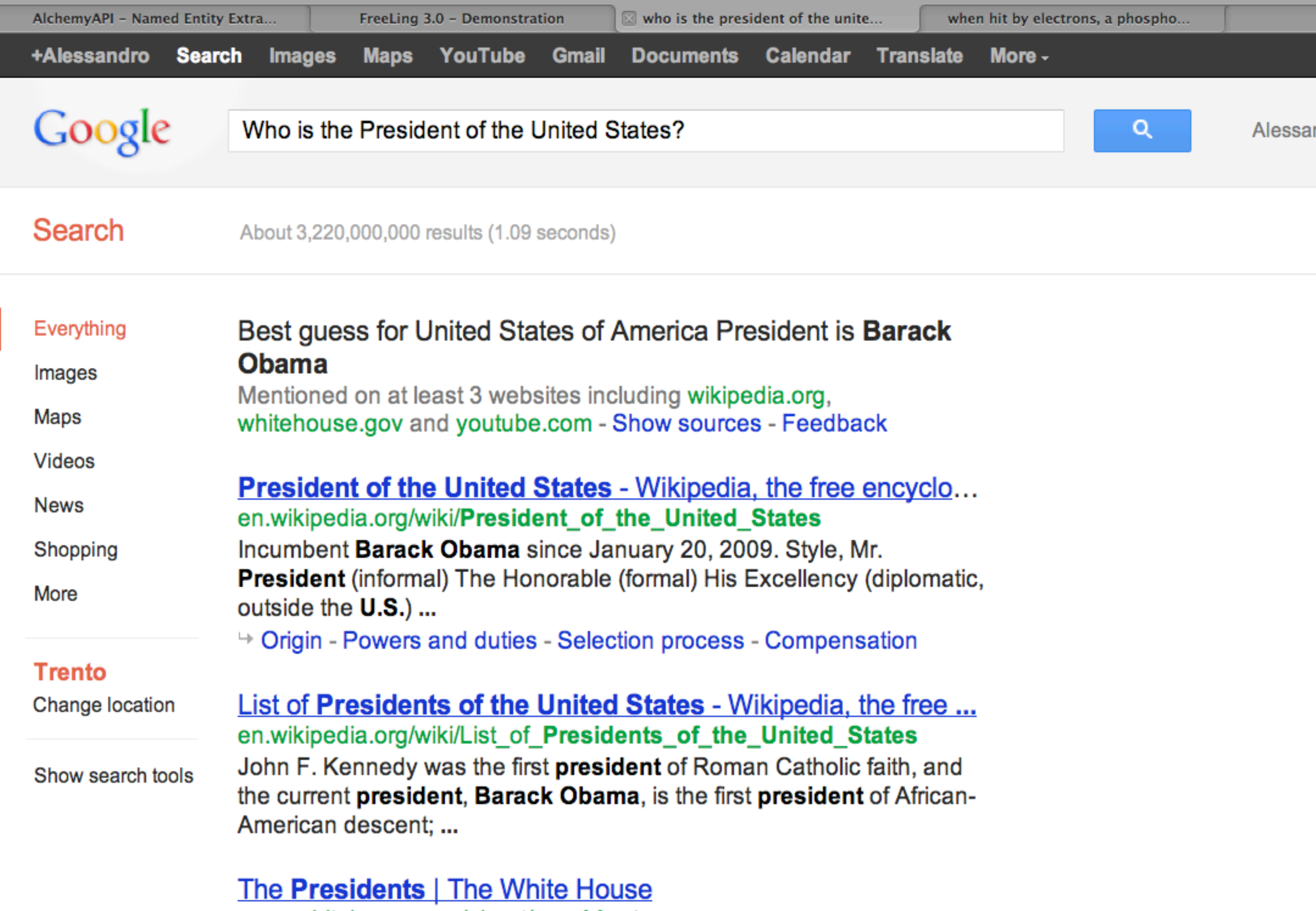
- Natural Language tools and techniques
  - Lemmatization
  - POS tagging
  - NER + gazetteer look up
  - Dependency and Constituency trees
  - Predicate Argument Structure
- Question Answering Pipeline
  - Similarity for supporting answers
  - QA tasks (open, restricted, factoid, non-factoid)

Motivations

- Approach to automatic Question Answering Systems
  1. Extract query keywords from the question
  2. Retrieve candidate passages containing such keywords (or synonyms)
  3. Select the most promising passage by means of query and answer similarity
- For example
  - Who is the President of the United States?

  (Yes) The president of the United States is Barack Obama

  (no) Glenn F. Tilton is President of the United Airlines


AlchemyAPI – Named Entity Extra...
FreeLing 3.0 – Demonstration
who is the president of the unite...
when hit by electrons, a phospho...
+Alessandro Search Images Maps YouTube Gmail Documents Calendar Translate More
Google
Who is the President of the United States?
Alessar
Search
About 3,220,000,000 results (1.09 seconds)
Everything
Images
Maps
Videos
News
Shopping
More
Trento
Change location
Show search tools
Best guess for United States of America President is Barack Obama
Mentioned on at least 3 websites including wikipedia.org, whitehouse.gov and youtube.com - Show sources - Feedback
President of the United States - Wikipedia, the free encyclo…
en.wikipedia.org/wiki/President_of_the_United_States
Incumbent Barack Obama since January 20, 2009. Style, Mr. President (informal) The Honorable (formal) His Excellency (diplomatic, outside the U.S.) ...
Origin - Powers and duties - Selection process - Compensation
List of Presidents of the United States - Wikipedia, the free ...
en.wikipedia.org/wiki/List_of_Presidents_of_the_United_States
John F. Kennedy was the first president of Roman Catholic faith, and the current president, Barack Obama, is the first president of African-American descent; ...
The Presidents | The White House

Motivations

- TREC has taught that this model is to weak
- Consider a more complex task, i.e. a Jeopardy cue
- *When hit by electrons, a phosphor gives off electromagnetic energy in this form*
  - Solutions: ***photons/light***
- What are the most similar fragments retrieved by a search engine?

ps Maps News Shopping Gmail more ▾

le

When hit by electrons, a phosphor gives off electromagnetic energy ✕

About 194,000 results (0.22 seconds) Advan

▶ Cathode-Ray Tube - body, used, chemical, characteristics, **form** ...

Sep 6, 2010 ... In order to **form** the **electron** beam into the correct shape, ... The actual conversion of electrical **energy** to light **energy** takes place on the ... For example, the **phosphor** known as yttrium oxide **gives off** a red glow ... complete explanation of electrostatic and **electromagnetic** focusing in the crt ...

www.scienceclarified.com › Ca-Ch - Cached - Similar

Beta particle - Wikipedia, the free encyclopedia

Beta particles are high-**energy**, high-speed **electrons** or positrons emitted by certain ... The beta particles emitted are a **form** of ionizing radiation also known as beta rays. ... by **electromagnetic** interactions and may **give off** bremsstrahlung x-rays. ... The well-known 'betalight' contains tritium and a **phosphor**. ...

en.wikipedia.org/wiki/Beta_particle - Cached - Similar

luminescence: Definition from Answers.com

Included on the **electromagnetic** spectrum are radio waves and microwaves; ... Though the Sun sends its **energy** to Earth in the **form** of light and heat from the .... Thanks to the **phosphor**, a fluorescent lamp **gives off** much more light than an ... The tube itself is coated

Motivations (2)

- This shows that:
  - Lexical similarity is not enough
  - Structure is required
- What kind of structures do we need?
- How to carry out structural similarity?

Information Retrieval Techniques

Indexing Unstructured Text

- Which plays of Shakespeare contain the words ***Brutus*** *AND* ***Caesar*** but *NOT* ***Calpurnia***?
- One could grep all of Shakespeare's plays for ***Brutus*** and ***Caesar,*** then strip out lines containing ***Calpurnia***?
  - Slow (for large corpora)
  - *NOT* ***Calpurnia*** is non-trivial
  - Other operations (e.g., find the word ***Romans*** near ***countrymen***) not feasible
  - Ranked retrieval (best documents to return)

Term-document incidence

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

1 if play contains word, 0 otherwise

***Brutus** AND **Caesar** but NOT*
Calpurnia

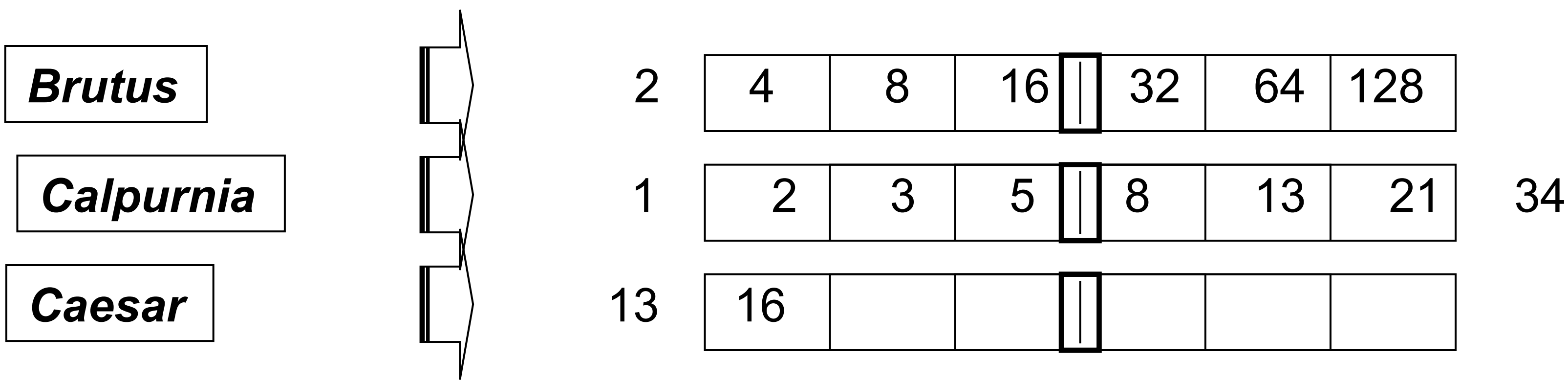
Incidence vectors

- So we have a 0/1 vector for each term.
- To answer query: take the vectors for ***Brutus, Caesar*** and ***Calpurnia*** (complemented) ➨ bitwise *AND*.
- 110100 *AND* 110111 *AND* 101111 = 100100.

Inverted index

For each term *T*, we must store a list of all documents that contain *T*.

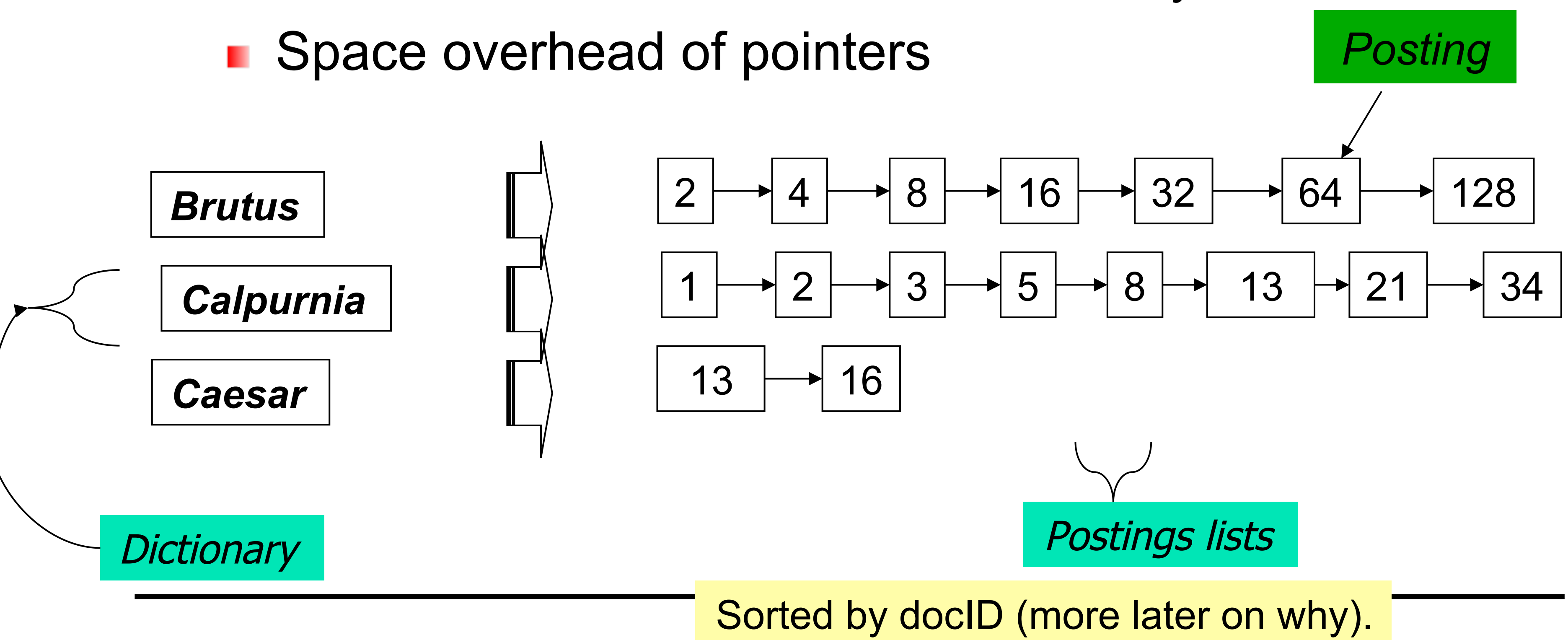
Do we use an array or a list for this?



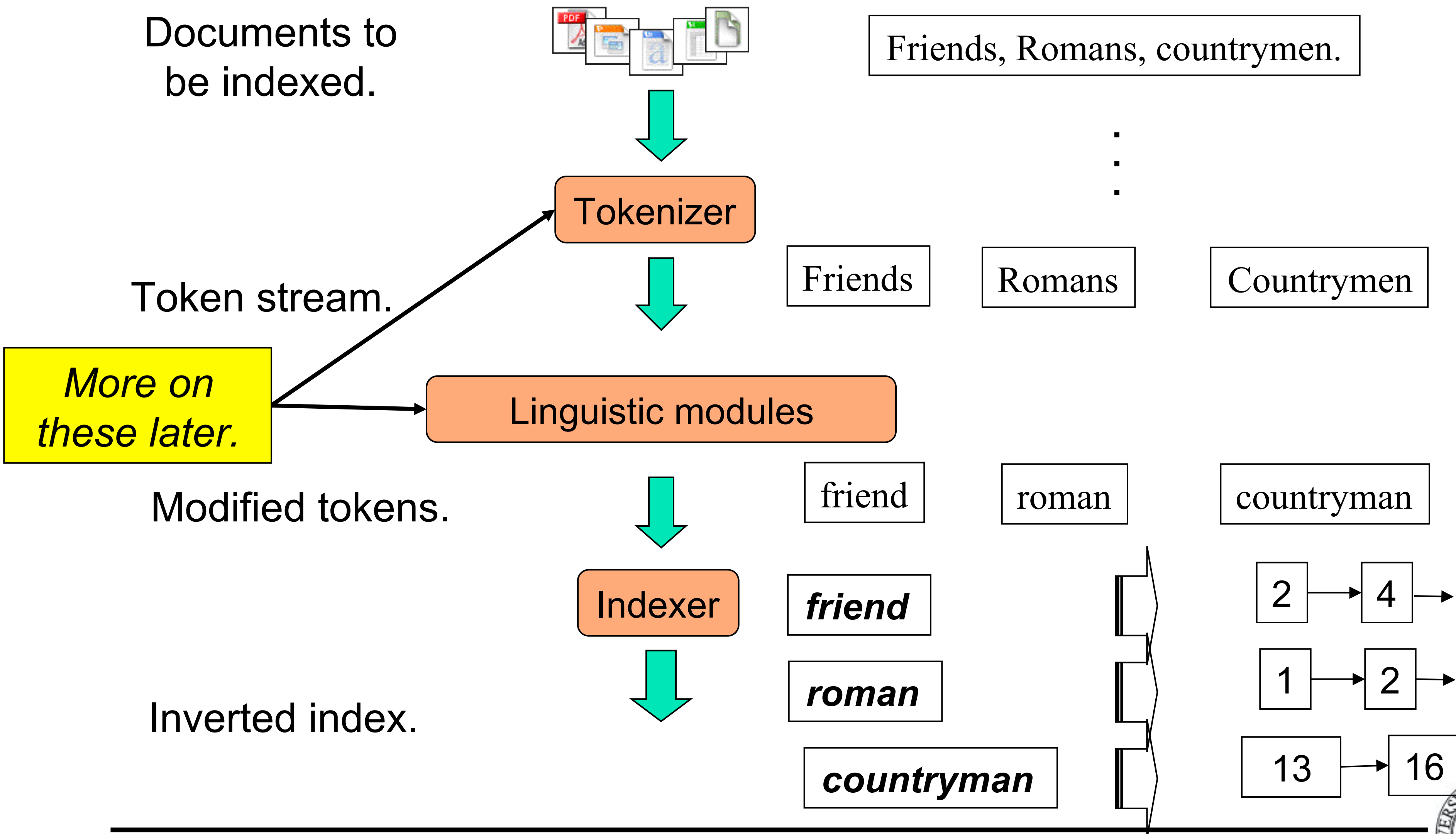
What happens if the word ***Caesar*** is added to document 14?

Inverted index

- Linked lists generally preferred to arrays
  - Dynamic space allocation
  - Insertion of terms into documents easy
  - Space overhead of pointers



Inverted index construction



Indexer steps

- Sequence of (Modified token, Document ID) pairs.

Doc 1

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

- Sort by terms.

Core indexing step.

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Boolean queries: Exact match

- The Boolean Retrieval model is being able to ask a query that is a Boolean expression:
  - Boolean Queries are queries using *AND, OR* and *NOT* to join query terms
    - Views each document as a set of words
    - Is precise: document matches condition or not.
- Primary commercial retrieval tool for 3 decades.
- Professional searchers (e.g., lawyers) still like Boolean queries:
  - You know exactly what you're getting.

Evidence accumulation

- 1 vs. 0 occurrence of a search term
  - 2 vs. 1 occurrence
  - 3 vs. 2 occurrences, etc.
  - Usually more seems better
- Need term frequency information in docs

Ranking search results

- Boolean queries give inclusion or exclusion of docs.
- Often we want to rank/group results
  - Need to measure proximity from query to each doc.
  - Need to decide whether docs presented to user are singletons, or a group of docs covering various aspects of the query.

IR vs. databases: Structured vs unstructured data

- Structured data tends to refer to information in "tables"

Employee	Manager	Salary
Smith	Jones	50000
Chang	Smith	60000
Ivy	Smith	50000

Typically allows numerical range and exact match (for text) queries, e.g.,

Salary < 60000 AND Manager = Smith.

Unstructured data

- Typically refers to free-form text
- Allows
  - Keyword queries including operators
  - More sophisticated "concept" queries, e.g.,
    - find all web pages dealing with *drug abuse*
- Classic model for searching text documents

Semi-structured data

- In fact almost no data is "unstructured"
- E.g., this slide has distinctly identified zones such as the *Title* and *Bullets*
- Facilitates "semi-structured" search such as
  - *Title* contains <u>data</u> AND *Bullets* contain <u>search</u>

… to say nothing of linguistic structure

From Binary term-document incidence matrix

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

Each document is represented by a binary vector $\in \{0,1\}^{|V|}$

To term-document count matrices

- Consider the number of occurrences of a term in a document:
  - Each document is a count vector in $\mathbb{N}^{v}$: a column below

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

Bag of words model

- Vector representation doesn't consider the ordering of words in a document
- *John is quicker than Mary* and *Mary is quicker than John* have the same vectors
- This is called the bag of words model.
- In a sense, this is a step back: The positional index was able to distinguish these two documents.

Term frequency tf

- The term frequency $\mathrm{tf}_{t,d}$ of term *t* in document *d* is defined as the number of times that *t* occurs in *d*.
- We want to use tf when computing query-document match scores. But how?
- Raw term frequency is not what we want:
  - A document with 10 occurrences of the term is more relevant than a document with 1 occurrence of the term.
  - But not 10 times more relevant.
- Relevance does not increase proportionally with term frequency.

NB: frequency = count in IR

Log-frequency weighting

- The log frequency weight of term t in d is

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

- 0 → 0, 1 → 1, 2 → 1.3, 10 → 2, 1000 → 4, etc.
- Score for a document-query pair: sum over terms *t* in both *q* and *d*:
- score $= \sum_{t \in q \cap d} (1 + \log \mathrm{tf}_{t,d})$
- The score is 0 if none of the query terms is present in the document.

Document frequency

- Rare terms are more informative than frequent terms
  - Recall stop words
- Consider a term in the query that is rare in the collection (e.g., *arachnocentric*)
- A document containing this term is very likely to be relevant to the query *arachnocentric*
- → We want a high weight for rare terms like *arachnocentric.*

idf weight

- df_t is the <u>document</u> frequency of *t*: the number of documents that contain *t*
  - df_t is an inverse measure of the informativeness of *t*
  - $\mathrm{df}_t \leq N$
- We define the idf (inverse document frequency) of *t* by

$$\mathrm{idf}_t = \log_{10}\,(N/\mathrm{df}_t)$$

  - We use log (N/df_t) instead of N/df_t to "dampen" the effect of idf.

Will turn out the base of the log is immaterial.

tf-idf weighting

- The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$w_{t,d} = \log(1+\mathrm{tf}_{t,d}) \times \log_{10}(N/\mathrm{df}_t)$$

- Best known weighting scheme in information retrieval
  - Note: the "-" in tf-idf is a hyphen, not a minus sign!
  - Alternative names: tf.idf, tf x idf
- Increases with the number of occurrences within a document
- Increases with the rarity of the term in the collection

Score for a document given a query

$$\text{Score}(q,d) = \sum_{t \in q \cap d} \text{tf} \times \text{idf}_{t,d}$$

- There are many variants
  - How "tf" is computed (with/without logs)
  - Whether the terms in the query are also weighted
  - …

Binary → count → weight matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Each document is now represented by a real-valued vector of tf-idf weights $\in \mathbb{R}^{|V|}$

Documents as vectors

- So we have a |V|-dimensional vector space
- Terms are axes of the space
- Documents are points or vectors in this space
- Very high-dimensional: tens of millions of dimensions when you apply this to a web search engine
- These are very sparse vectors - most entries are zero.

Queries as vectors

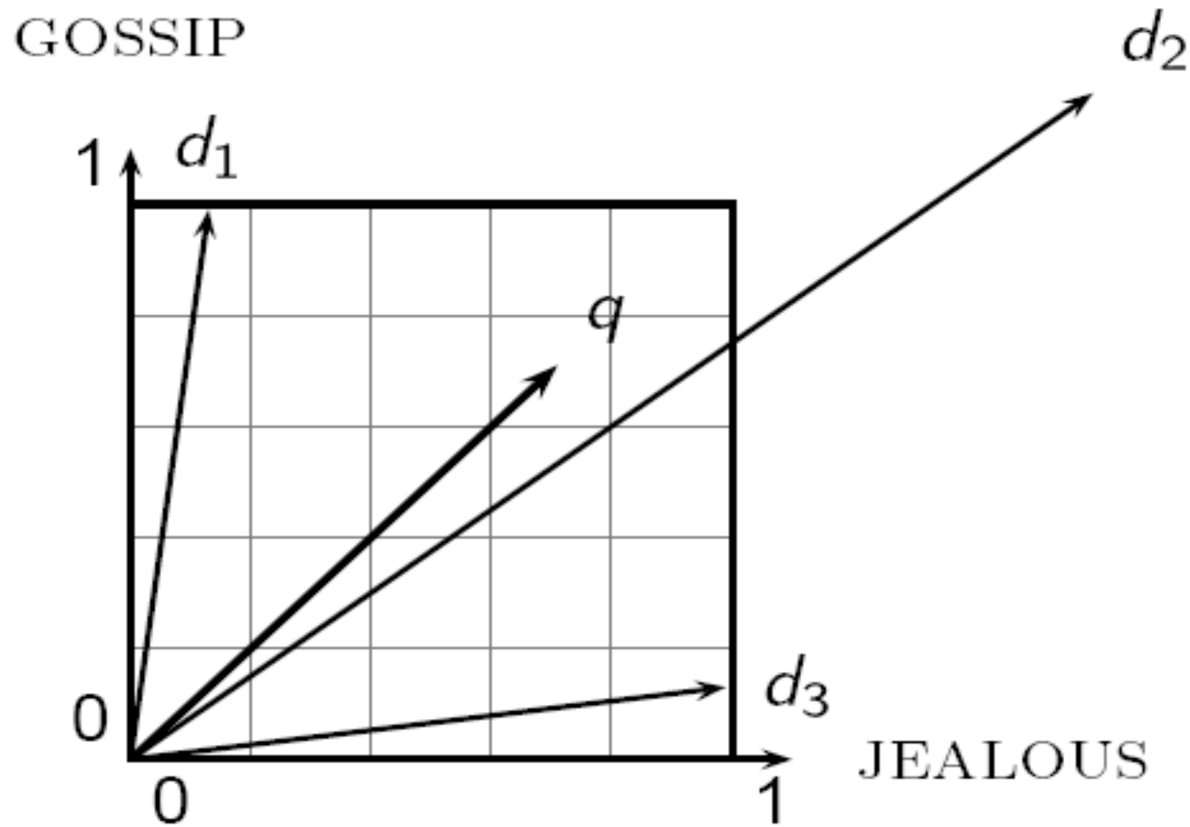
- Key idea 1: Do the same for queries: represent them as vectors in the space
- Key idea 2: Rank documents according to their proximity to the query in this space
- proximity = similarity of vectors
- proximity ≈ inverse of distance
- Instead: rank more relevant documents higher than less relevant documents

Formalizing vector space proximity

- First cut: distance between two points
  - (= distance between the end points of the two vectors)
- Euclidean distance?
- Euclidean distance is a bad idea . . .
- . . . because Euclidean distance is large for vectors of different lengths.

Why distance is a bad idea

The Euclidean distance between $\vec{q}$ and $\vec{d_2}$ is large even though the distribution of terms in the query $\vec{q}$ and the distribution of terms in the document $\vec{d_2}$ are very similar.



Use angle instead of distance

- Thought experiment: take a document *d* and append it to itself. Call this document *d*′.
- "Semantically" d and d′ have the same content
- The Euclidean distance between the two documents can be quite large
- The angle between the two documents is 0, corresponding to maximal similarity.

- Key idea: Rank documents according to angle with query.

From angles to cosines

- The following two notions are equivalent.
  - Rank documents in decreasing order of the angle between query and document
  - Rank documents in increasing order of cosine(query,document)
- Cosine is a monotonically decreasing function for the interval [0^o, 180^o]

Length normalization

- A vector can be (length-) normalized by dividing each of its components by its length – for this we use the L_2 norm:

$$\|\vec{x}\|_2 = \sqrt{\sum_i x_i^2}$$

- Dividing a vector by its L_2 norm makes it a unit (length) vector (on surface of unit hypersphere)
- Effect on the two documents d and d′ (d appended to itself) from earlier slide: they have identical vectors after length-normalization.
  - Long and short documents now have comparable weights

cosine(query,document)

Dot product

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\bullet\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|}\bullet\frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

qi is the tf-idf weight of term *i* in the query
di is the tf-idf weight of term *i* in the document

$\cos(\vec{q},\vec{d})$ is the cosine similarity of $\vec{q}$ and $\vec{d}$ … or, equivalently, the cosine of the angle between $\vec{q}$ and $\vec{d}$.

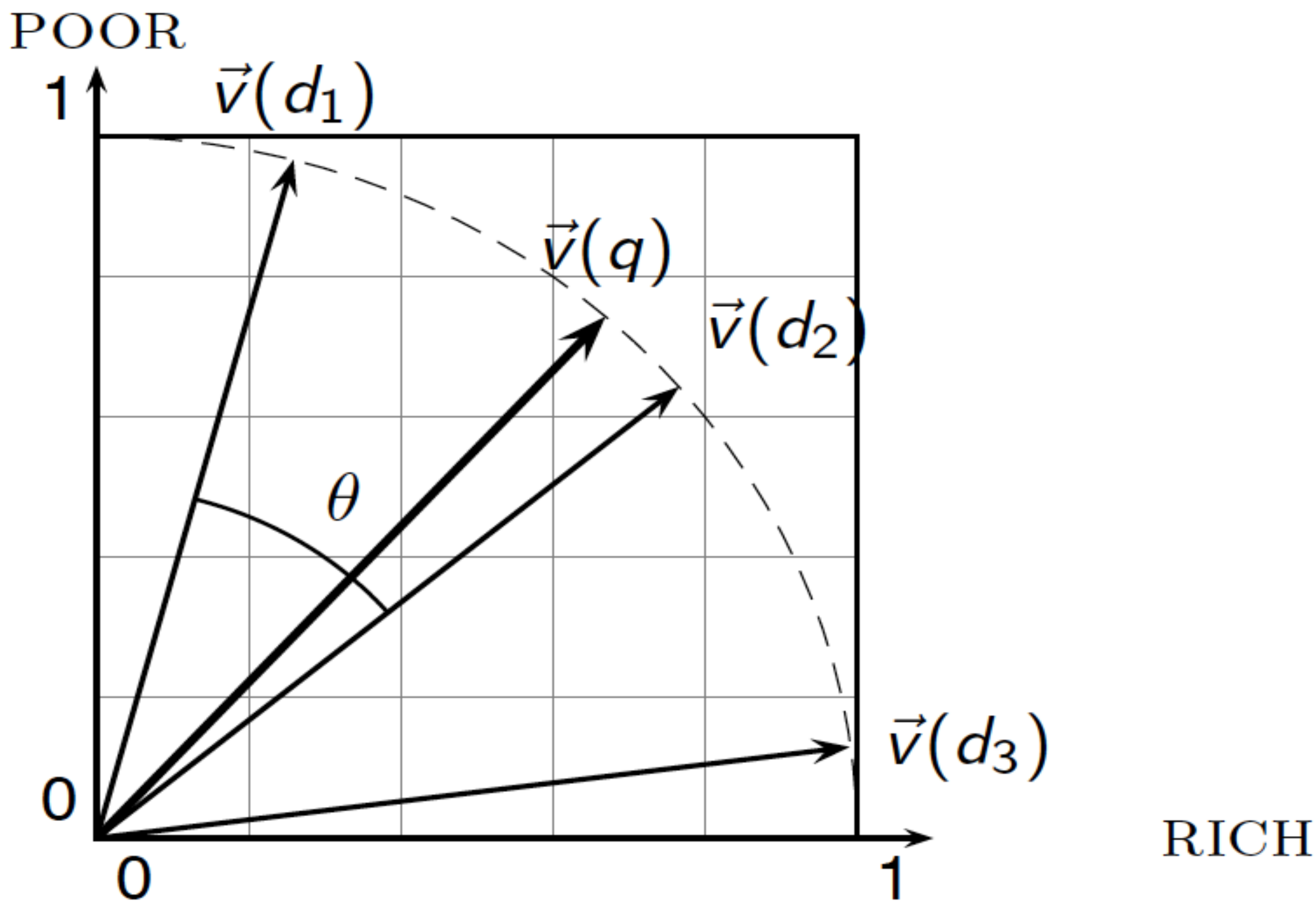
Cosine for length-normalized vectors

- For length-normalized vectors, cosine similarity is simply the dot product (or scalar product):

$$\cos(\vec{q},\vec{d}) = \vec{q} \bullet \vec{d} = \sum_{i=1}^{|V|} q_i d_i$$

for q, d length-normalized.

Cosine similarity illustrated



Performance Evaluation

Measures for a search engine

- We can quantify speed/size
- Quality of the retrieved documents
- Relevance measurement requires 3 elements:
    1. A benchmark document collection
    2. A benchmark suite of queries
    3. A usually binary assessment of either Relevant or Nonrelevant for each query and each document
        - Some work on more-than-binary, but not the standard

Evaluating an IR system

- Note: the **information need** is translated into a **query**
- Relevance is assessed relative to the **information need** *not* the **query**
- E.g., Information need: *I'm looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.*
- Query: ***wine red white heart attack effective***
- Evaluate whether the doc addresses the information need, not whether it has these words

Standard relevance benchmarks

- TREC - National Institute of Standards and Technology (NIST) has run a large IR test bed for many years
- Reuters and other benchmark doc collections used
- “Retrieval tasks” specified
  - sometimes as queries
- Human experts mark, for each query and for each doc, Relevant or Nonrelevant
  - or at least for subset of docs that some system returned for that query

Unranked retrieval evaluation: Precision and Recall

- **Precision**: fraction of retrieved docs that are relevant = P(relevant|retrieved)
- **Recall**: fraction of relevant docs that are retrieved = P(retrieved|relevant)

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

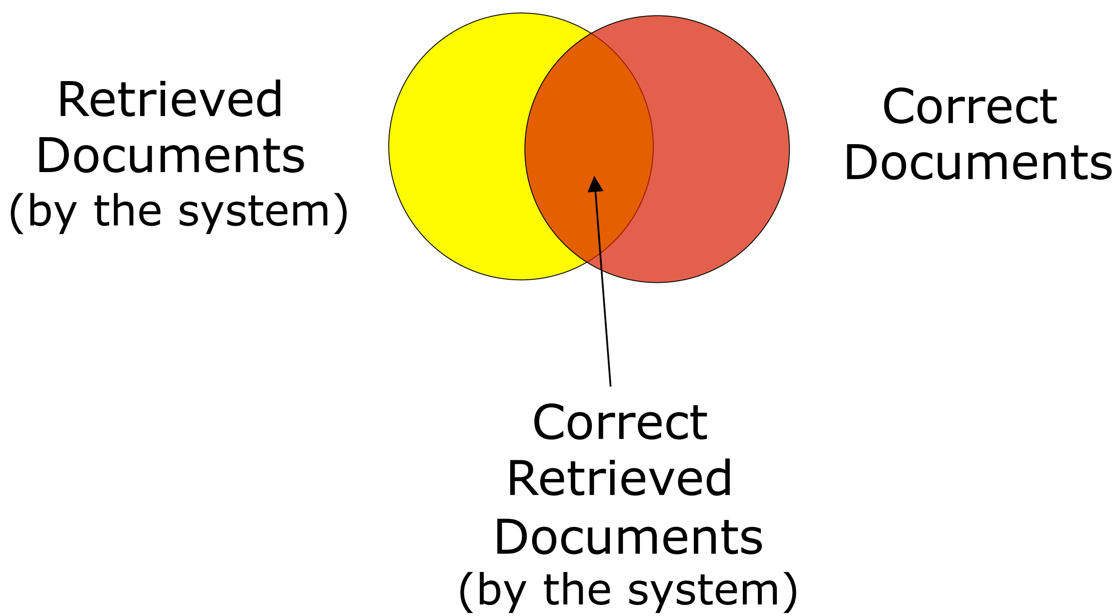
- Precision P = tp/(tp + fp)
- Recall R = tp/(tp + fn)

Should we instead use the accuracy measure for evaluation?

- Given a query, an engine classifies each doc as "Relevant" or "Nonrelevant"
- The **accuracy** of an engine: the fraction of these classifications that are correct
  - (tp + tn) / (tp + fp + fn + tn)
- **Accuracy** is a evaluation measure in often used in machine learning classification work
- Why is this not a very useful evaluation measure in IR?

Performance Measurements

- Given a set of document *T*
- Precision = # Correct Retrieved Document / # Retrieved Documents
- Recall = # Correct Retrieved Document/ # Correct Documents



Why not just use accuracy?

- How to build a 99.9999% accurate search engine on a low budget….

- People doing information retrieval *want to find something* and have a certain tolerance for junk.

Precision/Recall trade-off

- You can get high recall (but low precision) by retrieving all docs for all queries!
- Recall is a non-decreasing function of the number of docs retrieved

- In a good system, precision decreases as either the number of docs retrieved or recall increases
  - This is not a theorem, but a result with strong empirical confirmation

A combined measure: *F*

- Combined measure that assesses precision/recall tradeoff is **F measure** (weighted harmonic mean):

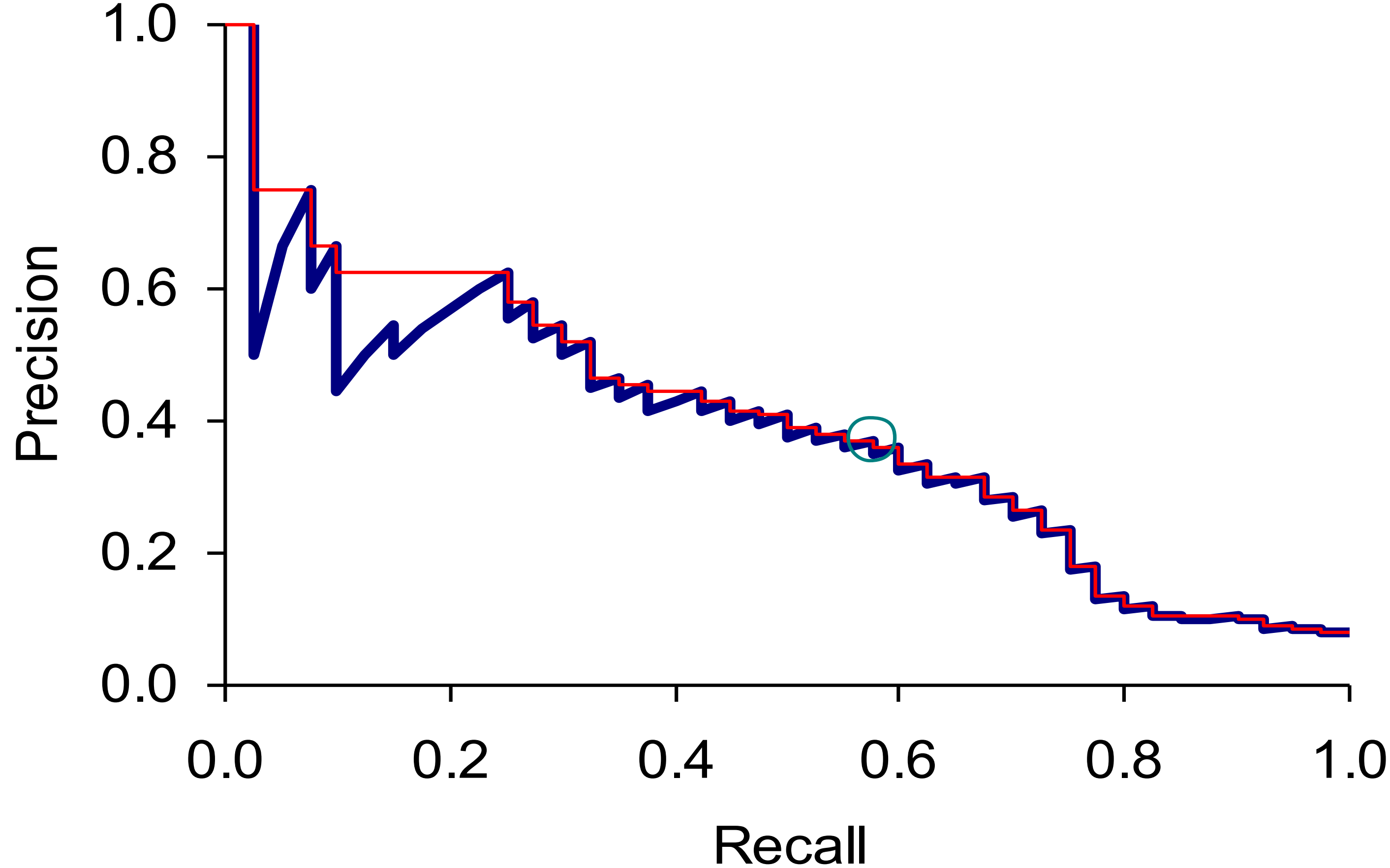
$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- People usually use balanced F_1 measure
  - i.e., with $\beta = 1$ or $\alpha = ½$
- Harmonic mean is a conservative average
  - See CJ van Rijsbergen, *Information Retrieval*

Evaluating ranked results

- Evaluation of ranked results:
  - The system can return any number of results
  - By taking various numbers of the top returned documents (levels of recall), the evaluator can produce a *precision-recall curve*

A precision-recall curve



Averaging over queries

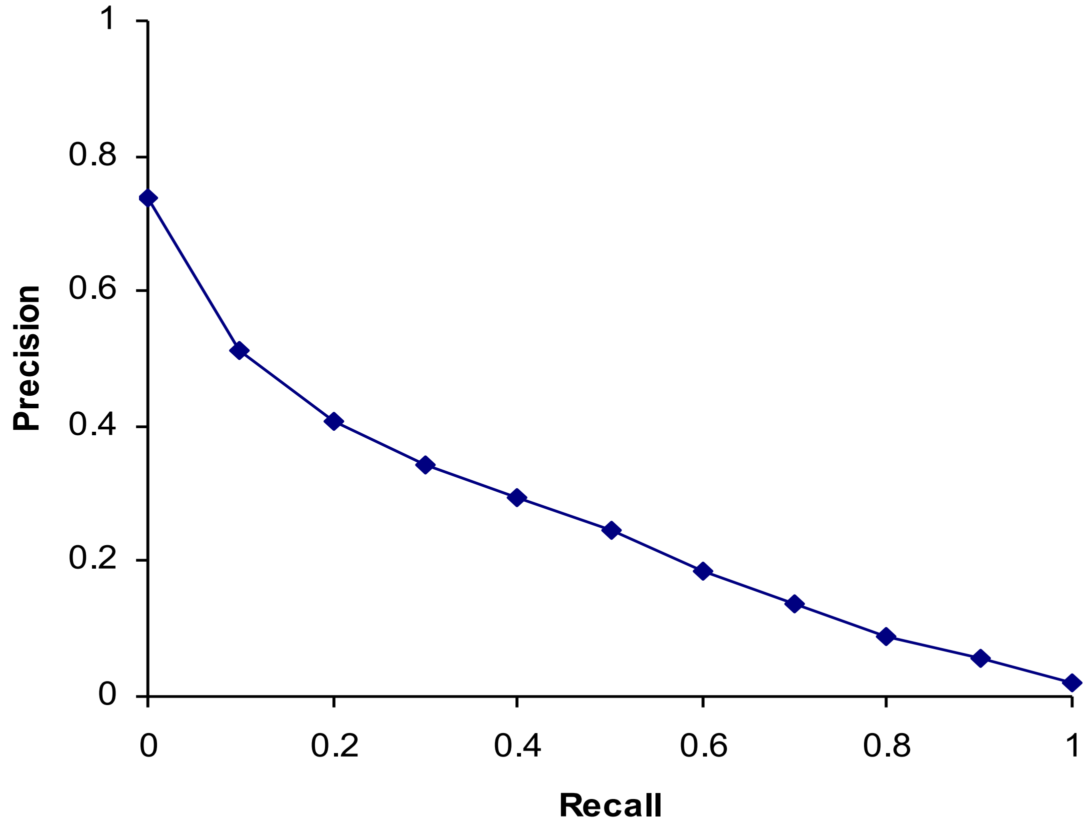
- A precision-recall graph for one query isn't a very sensible thing to look at
- You need to average performance over a whole bunch of queries.
- But there's a technical issue:
  - Precision-recall calculations place some points on the graph
  - How do you determine a value (interpolate) between the points?

Evaluation

- Graphs are good, but people want summary measures!
  - Precision at fixed retrieval level (**no CO**)
    - Precision-at-*k*: Precision of top *k* results
    - Perhaps appropriate for most of web search: all people want are good matches on the first one or two results pages
    - But: averages badly and has an arbitrary parameter of *k*
  - 11-point interpolated average precision (**CO**)
    - The standard measure in the early TREC competitions: you take the precision at 11 levels of recall varying from 0 to 1 by tenths of the documents, using interpolation (the value for 0 is always interpolated!), and average them
    - Evaluates performance at all recall levels

Typical (good) 11 point precisions

- SabIR/Cornell 8A1 11pt precision from TREC 8 (1999)



Yet more evaluation measures…

- Mean average precision (MAP) (**no CO**)
  - Average of the precision value obtained for the top *k* documents, each time a relevant doc is retrieved
  - Avoids interpolation, use of fixed recall levels
  - MAP for query collection is arithmetic ave.
    - Macro-averaging: each query counts equally
- R-precision (**no CO – just R relevant documents**)
  - If we have a known (though perhaps incomplete) set of relevant documents of size *Rel,* then calculate precision of the top *Rel* docs returned
  - Perfect system could score 1.0.

TREC

- TREC Ad Hoc task from first 8 TRECs is standard IR task
  - 50 detailed information needs a year
  - Human evaluation of pooled results returned
  - More recently other related things: Web track, HARD
- A TREC query (TREC 5)

  <top>

  <num> Number: 225

  <desc> Description:

  What is the main function of the Federal Emergency Management Agency (FEMA) and the funding level provided to meet emergencies? Also, what resources are available to FEMA such as people, equipment, facilities?

  </top>

Standard relevance benchmarks: Others

- GOV2
  - Another TREC/NIST collection
  - 25 million web pages
  - Largest collection that is easily available
  - But still 3 orders of magnitude smaller than what Google/Yahoo/MSN index
- NTCIR
  - East Asian language and cross-language information retrieval
- Cross Language Evaluation Forum (CLEF)
  - This evaluation series has concentrated on European languages and cross-language information retrieval.
- Many others

Text Categorization

Text Classification Problem

- Given:
  - a set of target categories: $C = \left\{ C^1, .., C^n \right\}$
  - the set *T* of documents,

define

$$f : T \rightarrow 2^C$$

- VSM (Salton89')
  - Features are dimensions of a Vector Space.
  - Documents and Categories are vectors of feature weights.
  - *d* is assigned to C^i if $\vec{d} \cdot \vec{C}^i > th$

The Vector Space Model

d_1: Politic

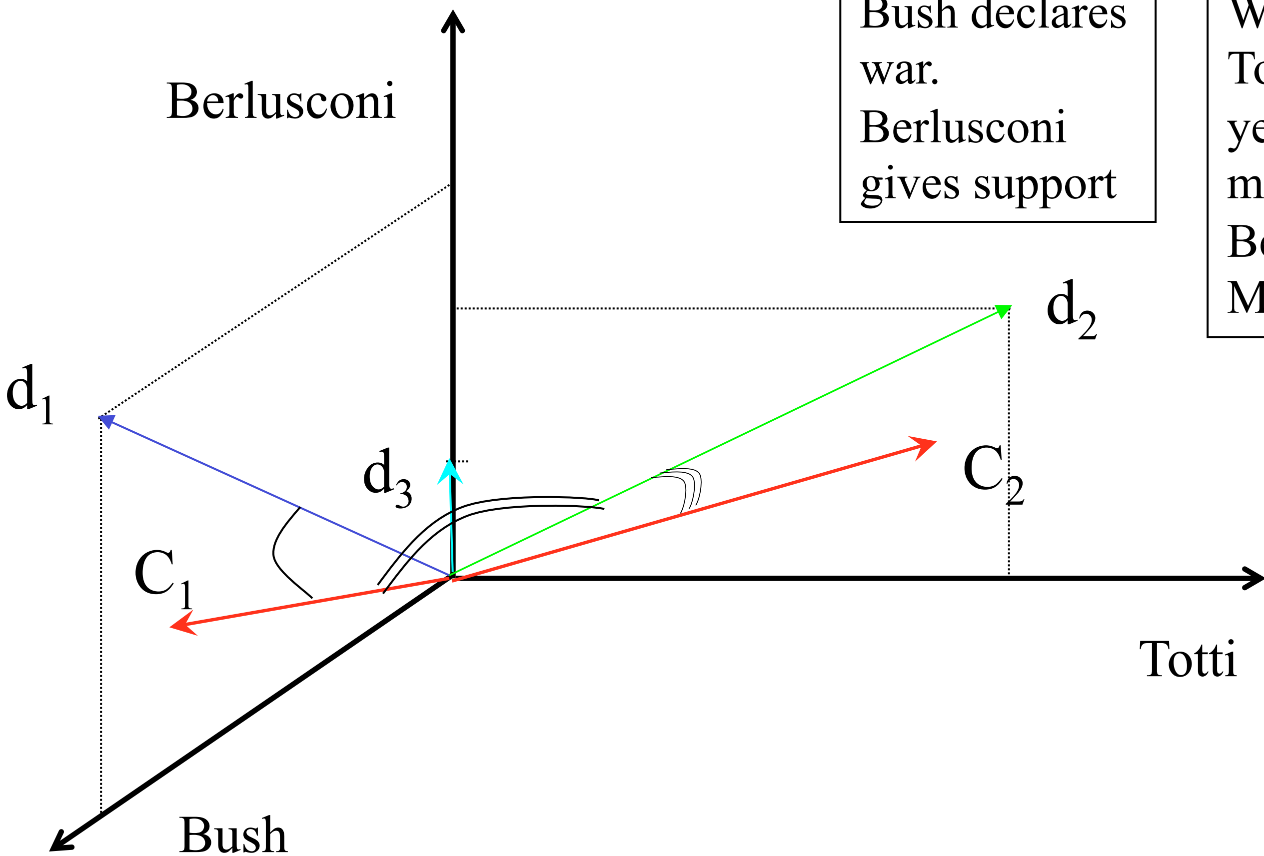
Bush declares war. Berlusconi gives support

d_2: Sport

Wonderful Totti in the yesterday match against Berlusconi's Milan

d_3:Economic

Berlusconi acquires Inzaghi before elections



C_1 : Politics Category

C_2 : Sport Category

Automated Text Categorization

- A corpus of pre-categorized documents
- Split document in two parts:
  - Training-set
  - Test-set
- Apply a supervised machine learning model to the training-set
  - Positive examples
  - Negative examples
- Measure the performances on the test-set
  - e.g., Precision and Recall

Feature Vectors

- Each example is associated with a vector of *n* feature types (e.g. unique words in TC)

$$\vec{x} = (0, \;..,1,..,0,..,0, \;..,1,..,0,..,0, \;..,1,..,0,..,0, \;..,1,..,0,.., \;1)$$

acquisition buy market sell stocks

- The dot product $\vec{x} \cdot \vec{z}$ counts the number of features in common

- This provides a sort of *similarity*

Text Categorization phases

- Corpus pre-processing (e.g. tokenization, stemming)
- Feature Selection (optionally)
  - Document Frequency, Information Gain, χ_2 , mutual information,...
- Feature weighting
  - for documents and profiles
- Similarity measure
  - between document and profile (e.g. scalar product)
- Statistical Inference
  - threshold application
- Performance Evaluation
  - Accuracy, Precision/Recall, BEP, f-measure,..

Feature Selection

- Some words, i.e. features, may be irrelevant
- For example, “function words” as: “the”, “on”,”those”…
- Two benefits:
  - efficiency
  - Sometime the accuracy
- Sort features by relevance and select the *m*-best

Statistical Quantity to sort feature

- Based on corpus counts of the pair

- A is the number of documents in which both f and c occur, i.e. (f, c);
- B is the number of documents in which only f occurs, i.e. $(f, \bar{c})$;
- C is the number of documents in which only c occurs, i.e. $(\bar{f}, c)$;
- D is the number of documents in which neither f nor c occur, i.e. $(\bar{f}, \bar{c})$;
- N is the total number of documents, i.e. $A + B + C + D$.

Statistical Selectors

- Chi-square, Pointwise MI and MI

$$\chi^2(f,c) = \frac{N \times (AD - CB)^2}{(A+C)(B+D)(A+B)(C+D)}$$

$$PMI(f,c) = log\frac{P(f,c)}{P(f) \times P(c)}$$

$$MI(f) \quad (f,C) = -\sum_{c \in \mathcal{C}} P(c)log(P(c)) + P(f)\sum_{c \in \mathcal{C}} P(c|f)log(P(c|f)) + P(\bar{f})\sum_{c \in \mathcal{C}} P(c|\bar{f})log(P(c|\bar{f}))$$

Profile Weighting: the Rocchio's formula

- ω_f^d, the weight of *f* in *d*
  - Several weighting schemes (e.g. TF * IDF, Salton 91')

- $\vec{C}_f^i$, the profile weights of *f* in C_i:

$$\vec{C}_f^i = \max\left\{ 0, \frac{\beta}{|T_i|} \sum_{d \in T_i} \omega_f^d - \frac{\gamma}{|\bar{T}_i|} \sum_{d \in \bar{T}_i} \omega_f^d \right\}$$

- T_i, the training documents in C^i

Similarity estimation

- Given the document and the category representation

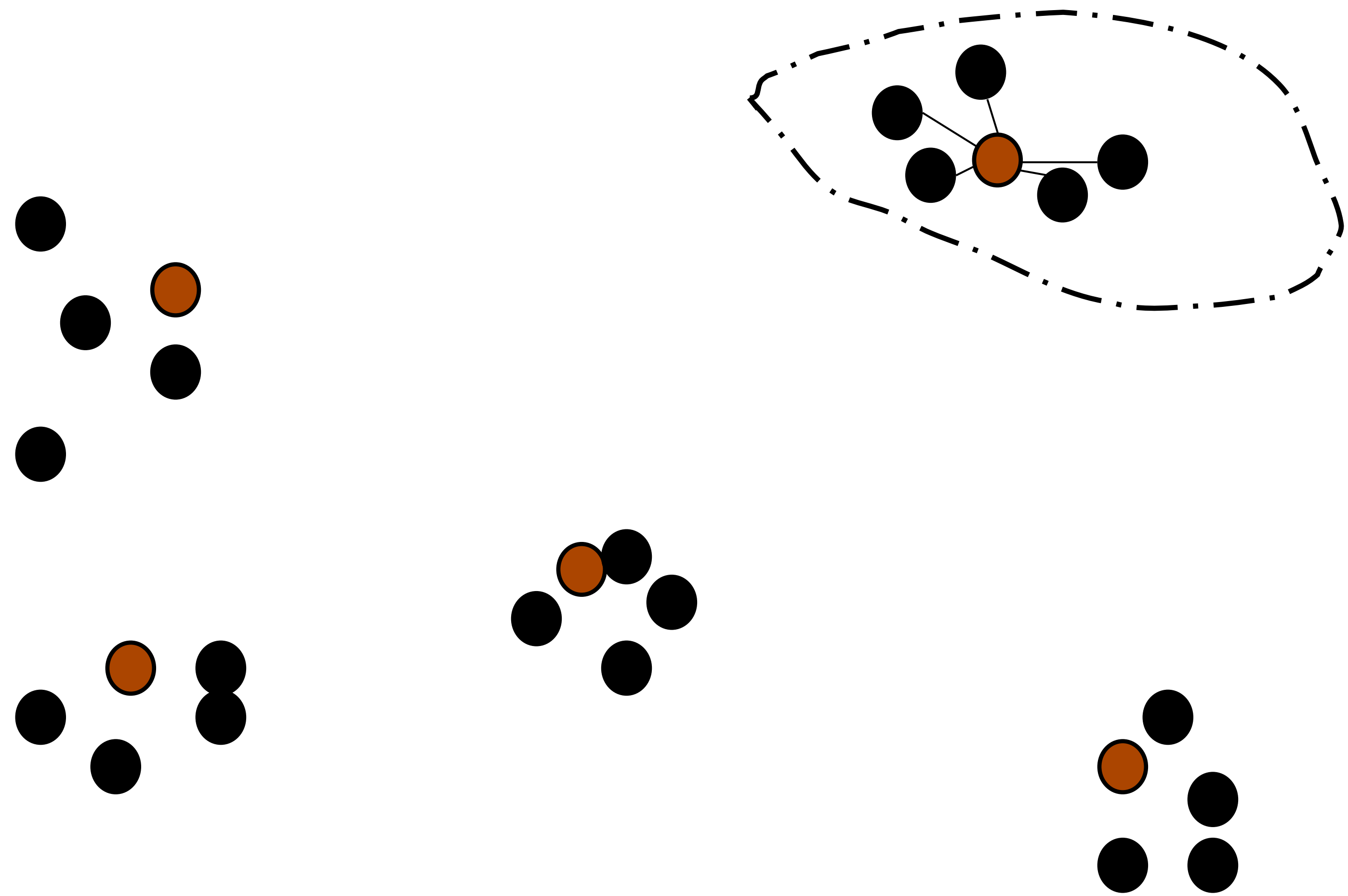
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- It can be defined the following similarity function (cosine measure

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

- *d* is assigned to C^i if $\vec{d} \cdot \vec{C}^i > \sigma$

Clustering



Experiments

- Reuters Collection 21578 Apté split (Apté94)
  - 90 classes (12,902 docs)
  - A fixed splitting between training and test set
  - 9603 vs 3299 documents
- Tokens
  - about 30,000 different
- Other different versions have been used but …

  most of TC results relate to the 21578 Apté
  - [Joachims 1998], [Lam and Ho 1998], [Dumais et al. 1998], [Li Yamanishi 1999], [Weiss et al. 1999],

    [Cohen and Singer 1999]…

A Reuters document- Acquisition Category

CRA SOLD FORREST GOLD FOR 76 MLN DLRS - WHIM CREEK

SYDNEY, April 8 - <Whim Creek Consolidated NL> said the consortium it is leading will pay 76.55 mln dlrs for the acquisition of CRA Ltd's <CRAA.S> <Forrest Gold Pty Ltd> unit, reported yesterday.

CRA and Whim Creek did not disclose the price yesterday.

Whim Creek will hold 44 pct of the consortium, while <Austwhim Resources NL> will hold 27 pct and <Croesus Mining NL> 29 pct, it said in a statement.

As reported, Forrest Gold owns two mines in Western Australia producing a combined 37,000 ounces of gold a year. It also owns an undeveloped gold project.

A Reuters document- Crude-Oil Category

FTC URGES VETO OF GEORGIA GASOLINE STATION BILL

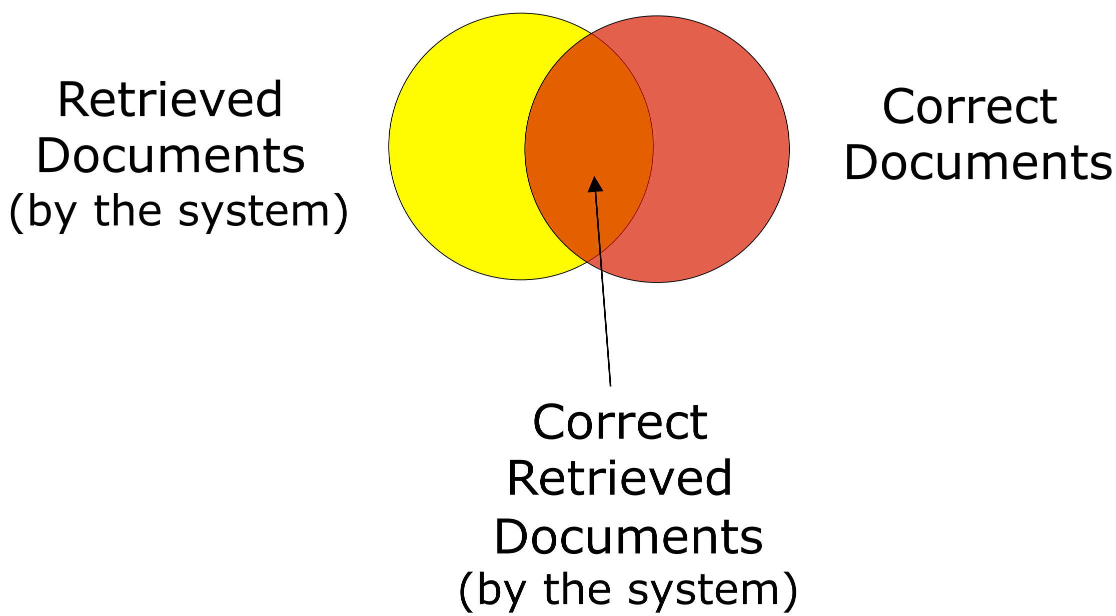
WASHINGTON, March 20 - The Federal Trade Commission said its staff has urged the governor of Georgia to veto a bill that would prohibit petroleum refiners from owning and operating retail gasoline stations.

The proposed legislation is aimed at preventing large oil refiners and marketers from using predatory or monopolistic practices against franchised dealers.

But the FTC said fears of refiner-owned stations as part of a scheme of predatory or monopolistic practices are unfounded. It called the bill anticompetitive and warned that it would force higher gasoline prices for Georgia motorists.

Performance Measurements

- Given a set of document *T*
- Precision = # Correct Retrieved Document / # Retrieved Documents
- Recall = # Correct Retrieved Document/ # Correct Documents



Precision and Recall of C_i

- a, corrects
- b, mistakes

The *Precision* and *Recall* are defined by the above counts:

$$Precision_i = \frac{a_i}{a_i + b_i}$$

$$Recall_i = \frac{a_i}{a_i + c_i}$$

Performance Measurements (cont'd)

- Breakeven Point
  - Find thresholds for which

    Recall = Precision
  - Interpolation
- f-measure
  - Harmonic mean between precision and recall
- Global performance on more than two categories
  - Micro-average
    - The counts refer to classifiers
  - Macro-average (average measures over all categories)

F-measure e MicroAverages

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

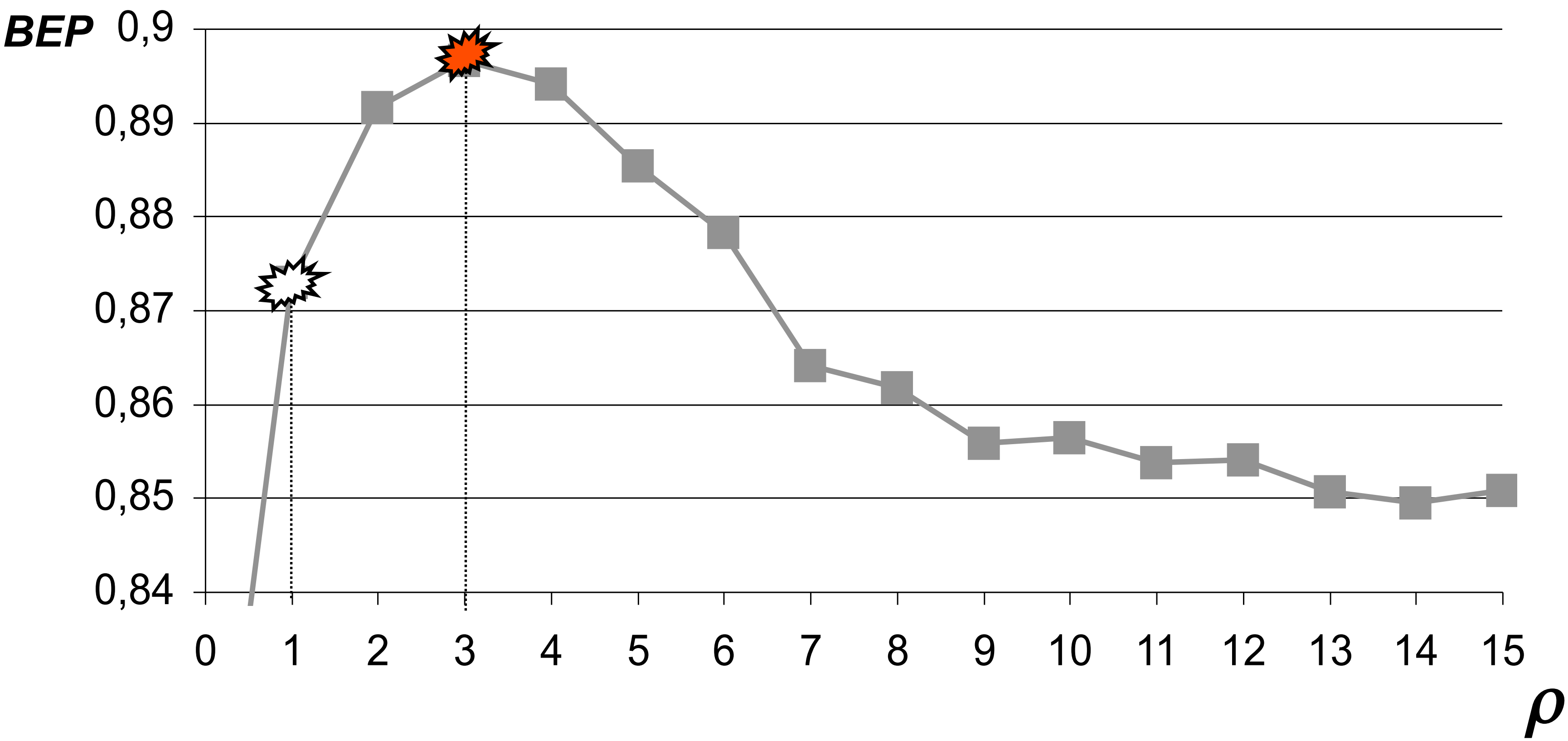
$$\mu Precision = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + b_i}$$

$$\mu Recall = \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i + c_i}$$

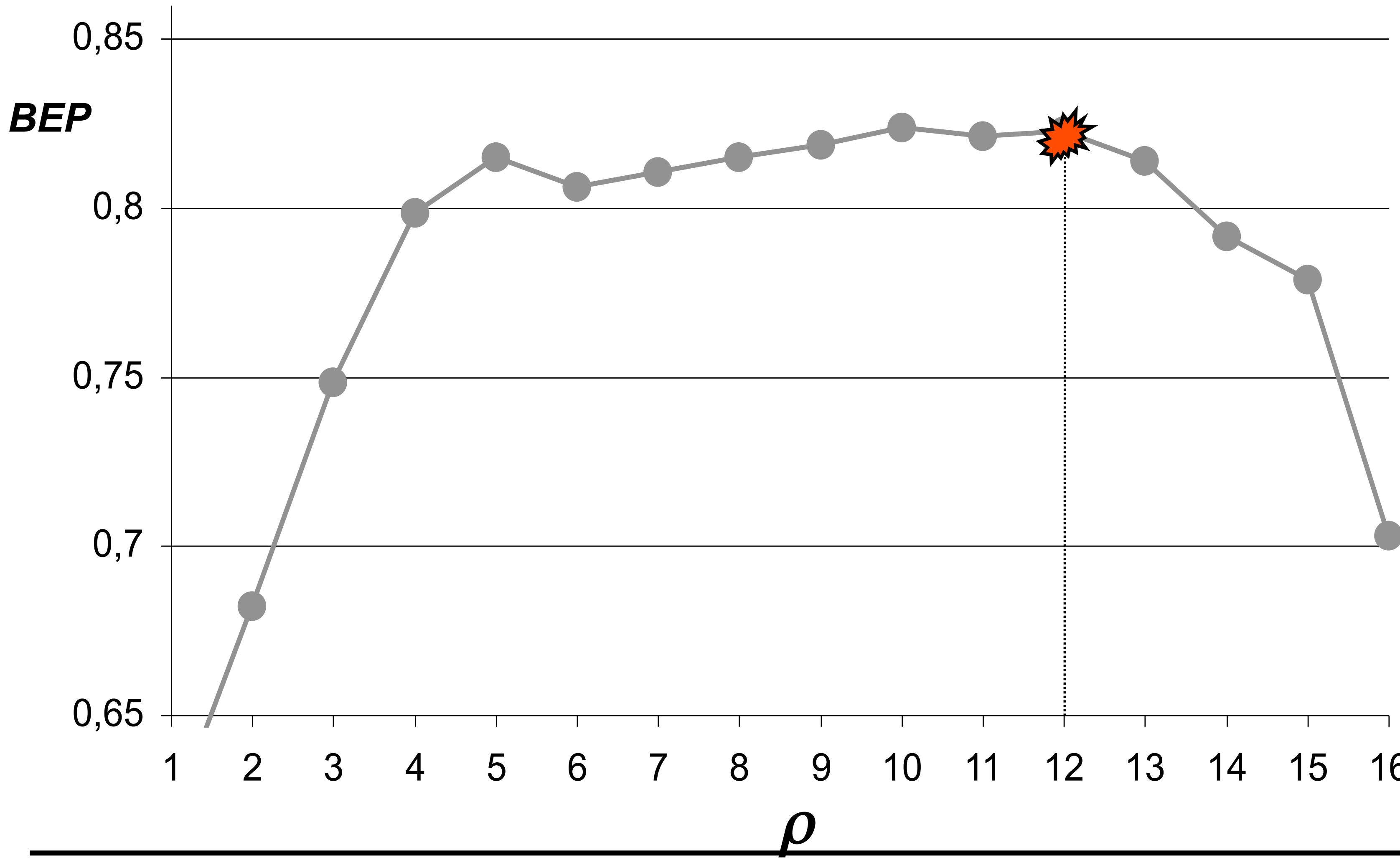
$$\mu BEP = \frac{\mu Precision + \mu Recall}{2}$$

$$\mu f_1 = \frac{2 \times \mu Precision \times \mu Recall}{\mu Precision + \mu Recall}$$

The Impact of ρ parameter on Acquisition category



The impact of ρ parameter on Trade category



N-fold cross validation

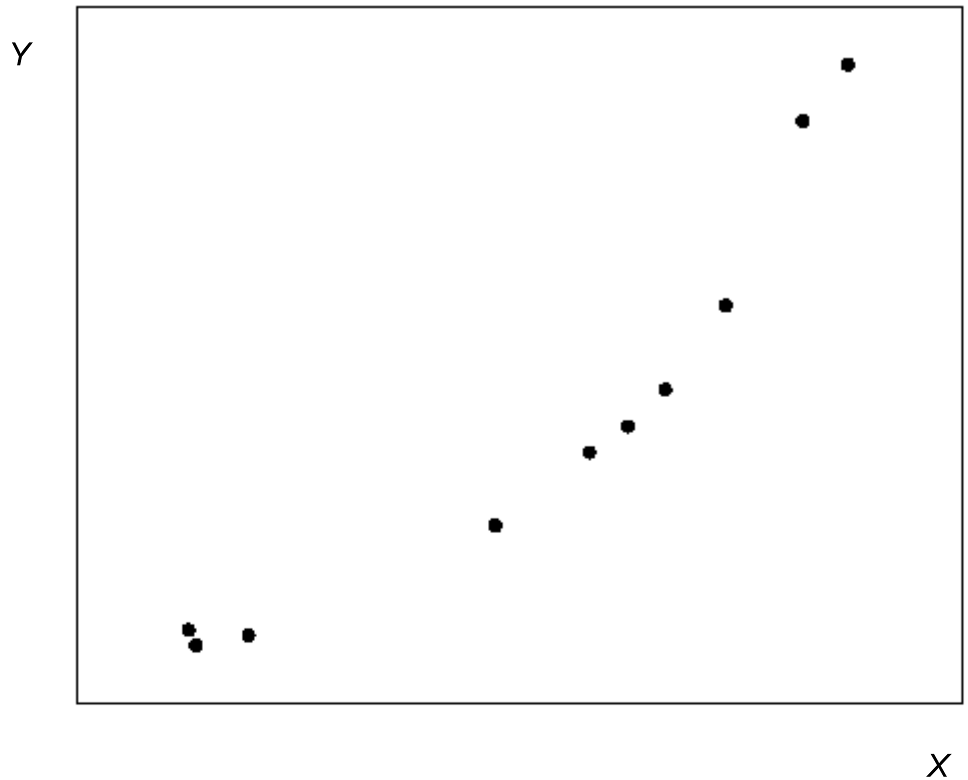
- Divide training set in *n* parts
  - One is used for testing
  - *n-1* for training
- This can be repeated *n* times for *n* distinct test sets
- Average and Std. Dev. are the final performance index

Classification, Ranking, Regression and Multiclassification

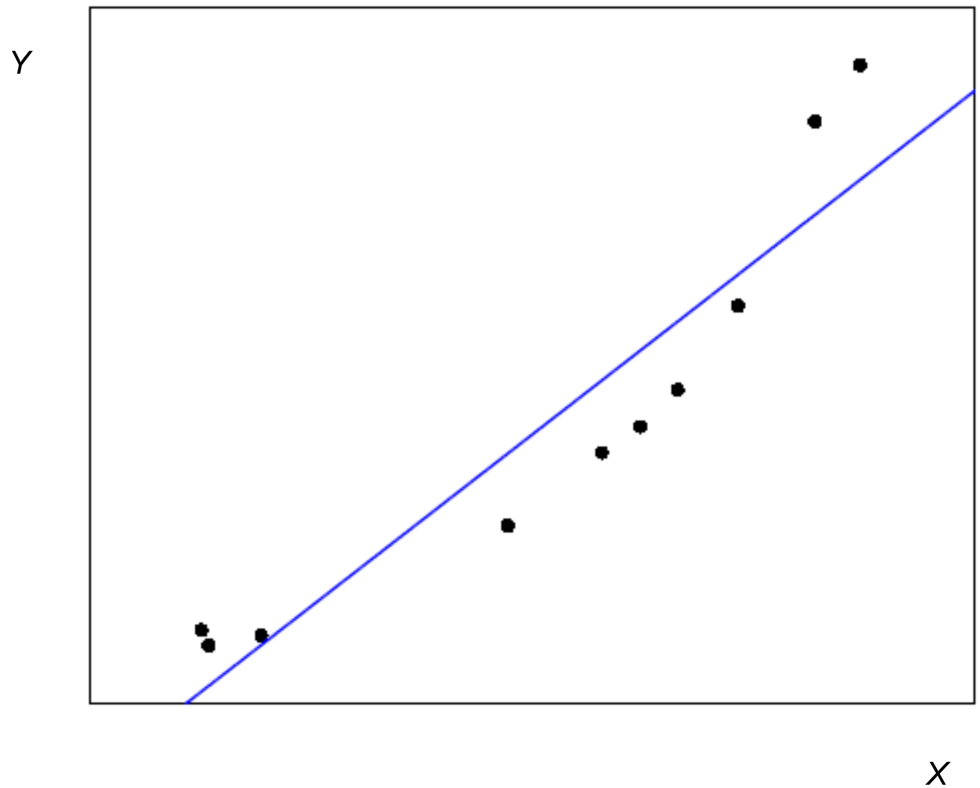
What is Statistical Learning?

- Statistical Methods – Algorithms that learn relations in the data from examples
- Simple relations are expressed by pairs of variables: $\langle x_1,y_1\rangle, \langle x_2,y_2\rangle,\ldots, \langle x_n,y_n\rangle$
- Learning f such that evaluate y^* given a new value x^*, i.e. $\langle x^*, f(x^*)\rangle = \langle x^*, y^*\rangle$

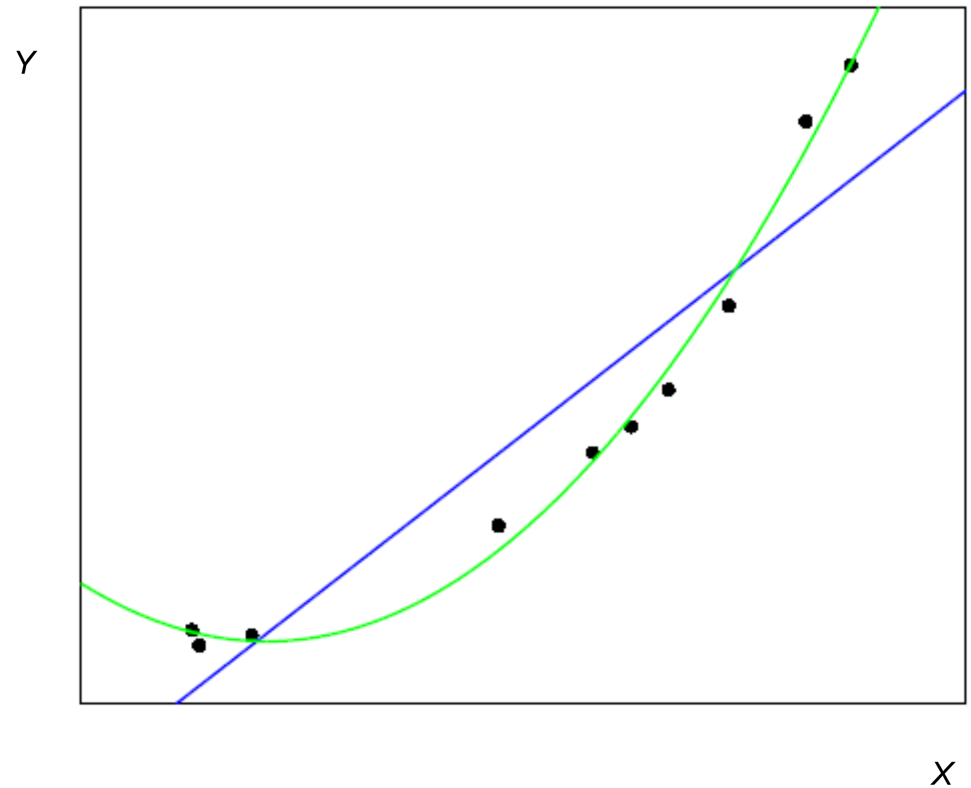
You have already tackled the learning problem



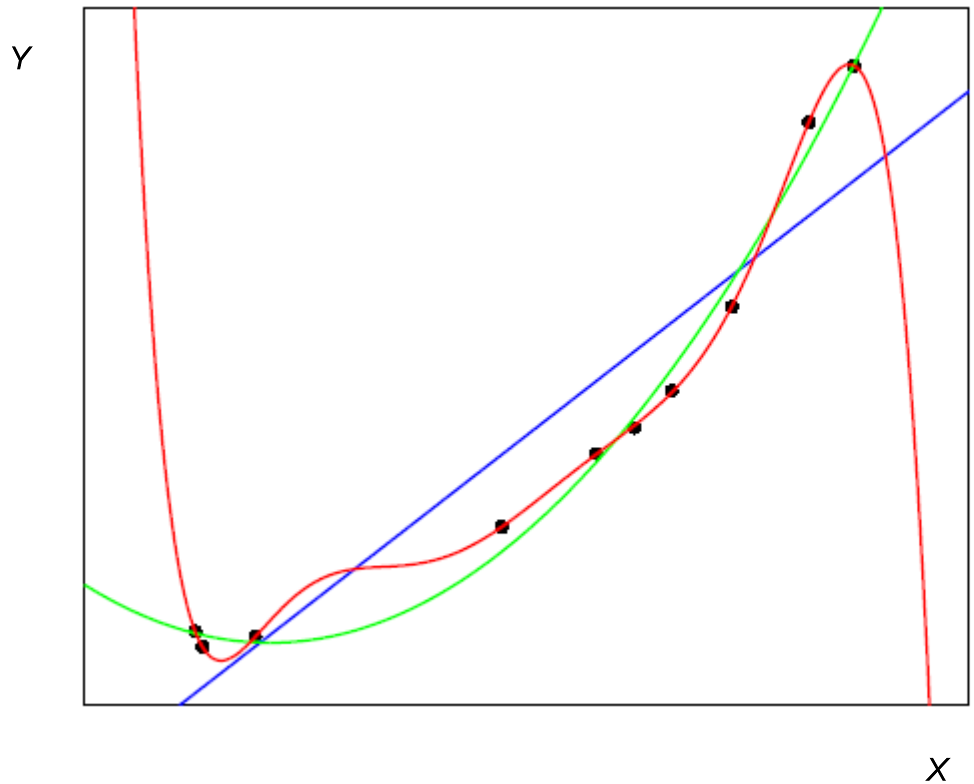
Linear Regression



Degree 2



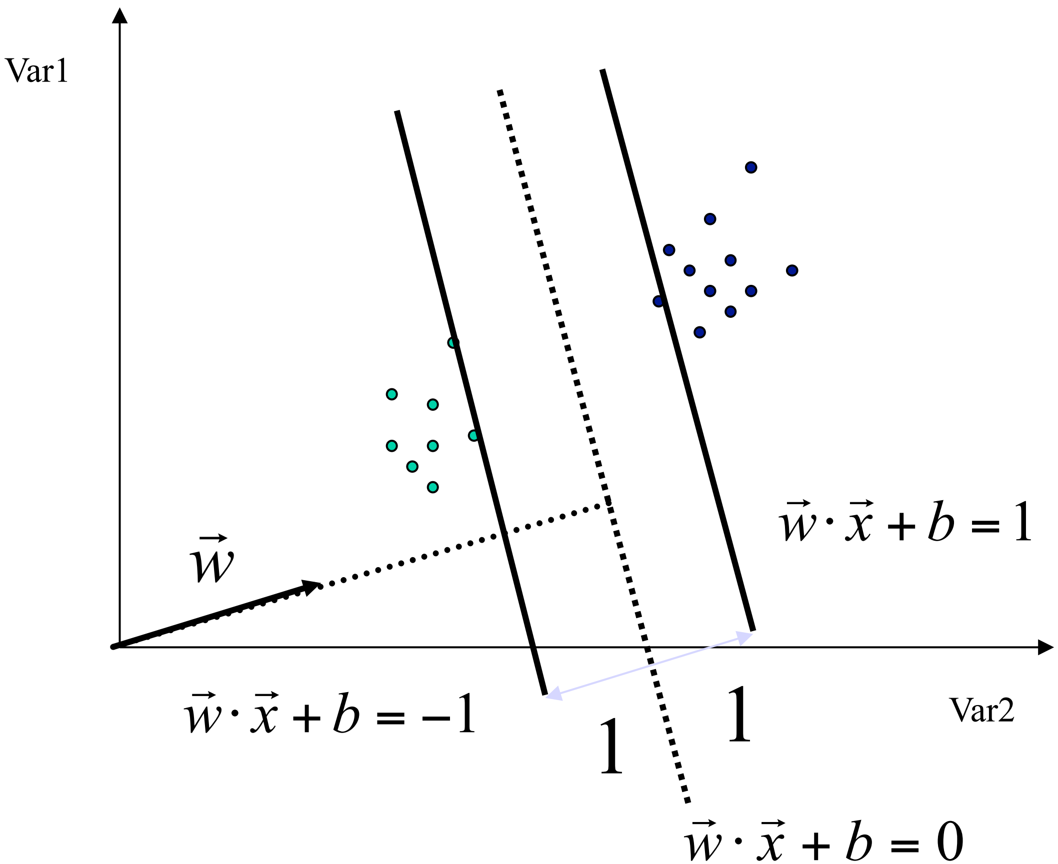
Degree



Machine Learning Problems

- Overfitting
- How dealing with millions of variables instead of only two?
- How dealing with real world objects instead of real values?

Support Vector Machines



There is a scale for which *k=1*.

The problem transforms in:

$$\max \frac{2}{\|\vec{w}\|}$$

$$\vec{w}\cdot\vec{x}+b \geq +1, \text{ if } \vec{x} \text{ is positive}$$

$$\vec{w}\cdot\vec{x}+b \leq -1, \text{ if } \vec{x} \text{ is negative}$$

The Ranking SVM
[Herbrich et al. 1999, 2000; Joachims et al. 2002]

- The aim is to classify instance pairs as correctly ranked or incorrectly ranked
  - This turns an ordinal regression problem back into a binary classification problem
- We want a ranking function f such that

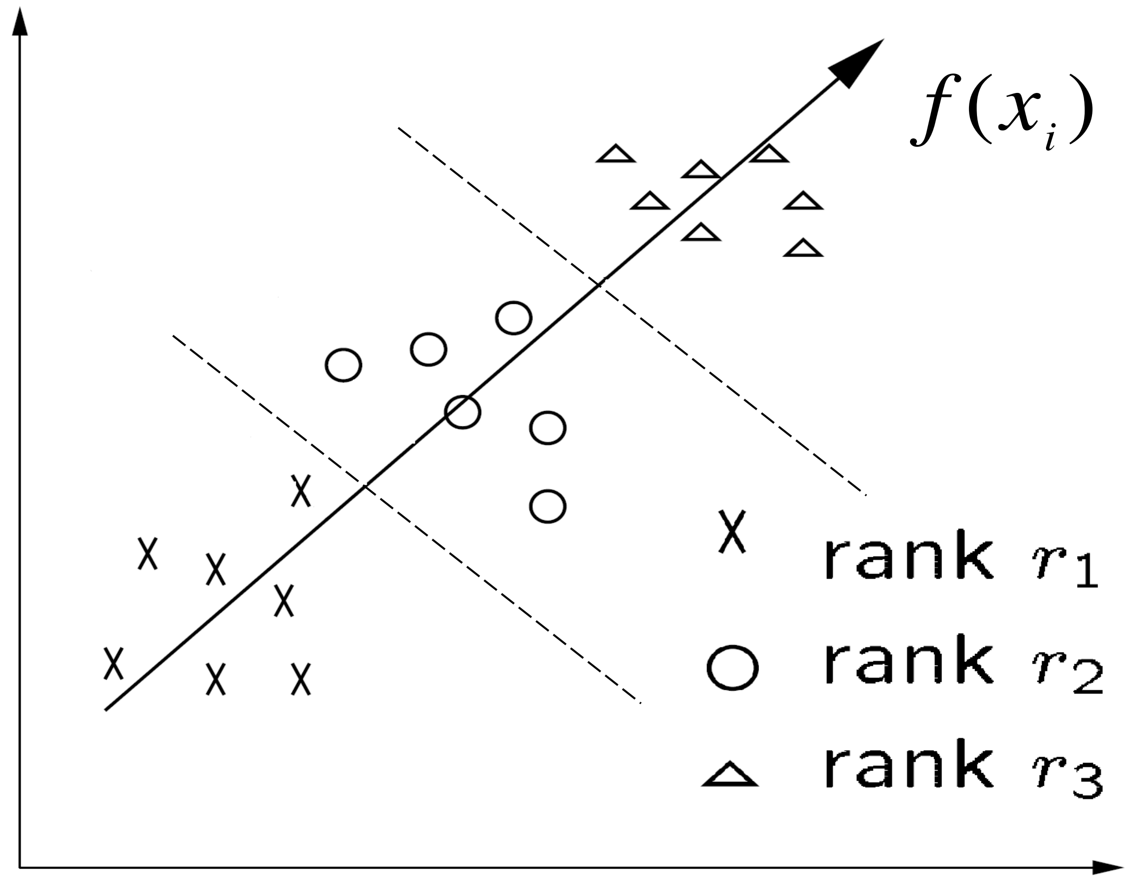
$$\boldsymbol{x}_i > \boldsymbol{x}_j \text{ iff } f(\boldsymbol{x}_i) > f(\boldsymbol{x}_j)$$

- … or at least one that tries to do this with minimal error
- Suppose that f is a linear function

$$f(\boldsymbol{x}_i) = \mathbf{w} \bullet \boldsymbol{x}_i$$

The Ranking SVM

- Ranking Model: $f(\boldsymbol{x}_i)$



The Ranking SVM

- Then (combining the two equations on the last slide):

$$\boldsymbol{x}_i > \boldsymbol{x}_j \text{ iff } \mathbf{w}\bullet\boldsymbol{x}_i - \mathbf{w}\bullet\boldsymbol{x}_j > 0$$

$$\boldsymbol{x}_i > \boldsymbol{x}_j \text{ iff } \mathbf{w}\bullet(\boldsymbol{x}_i - \boldsymbol{x}_j) > 0$$

- Let us then create a new instance space from such pairs:

$$z_k = x_i - x_k$$

$$y_k = +1, -1 \text{ as } \boldsymbol{x}_i \geq , < \boldsymbol{x}_k$$

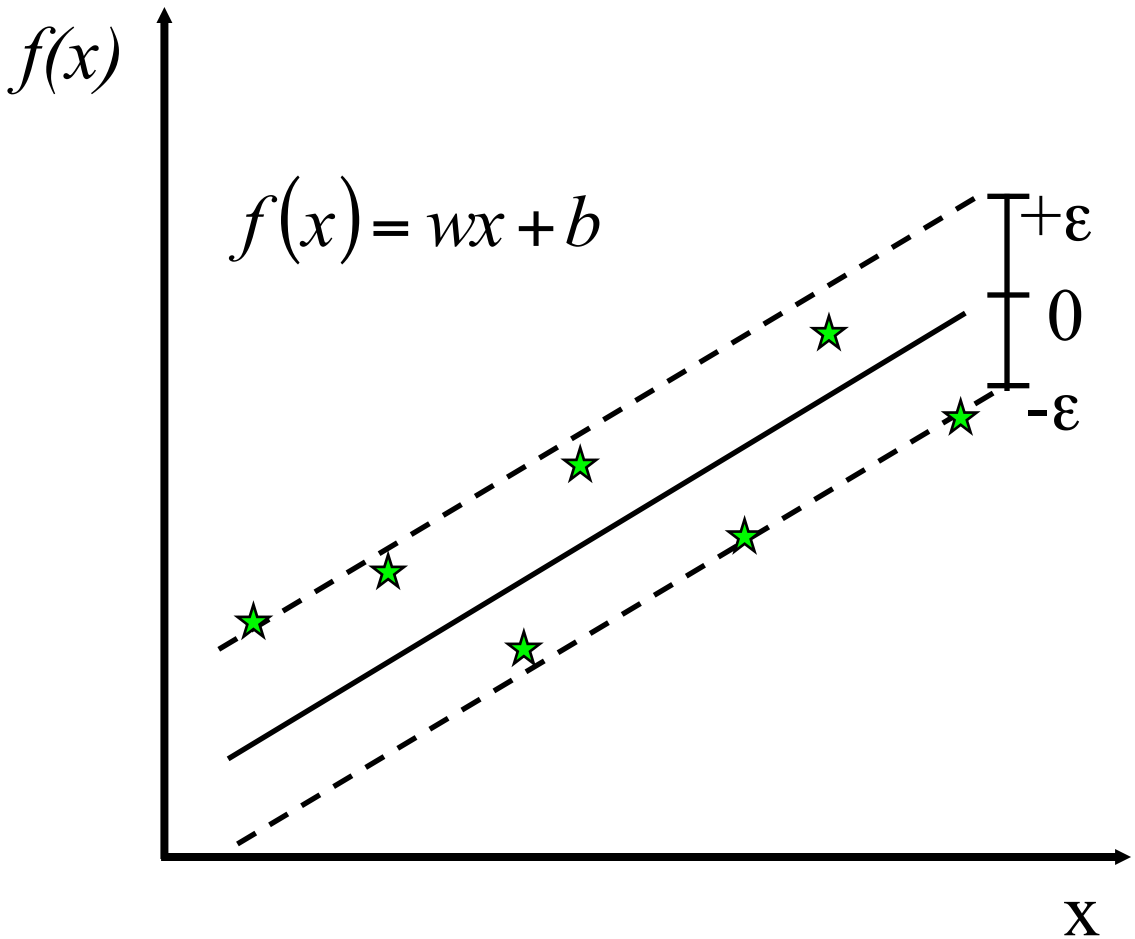
Support Vector Ranking

$$\begin{cases} min \quad \frac{1}{2}||\vec{w}|| + C\sum_{i=1}^{m}\xi_i^2 \\ y_k(\vec{w}\cdot(\vec{x_i}-\vec{x_j})+b) \geq 1-\xi_k, \quad \forall i,j=1,..,m \\ \xi_k \geq 0, \quad k=1,..,m^2 \end{cases}$$

y_k = 1 if $rank(\vec{x_i}) > rank(\vec{x_j})$,–1 otherwise, where $k = i \times m + j$

- Given two examples we build one example (x_i , x_j)

Support Vector Regression (SVR)



Solution:

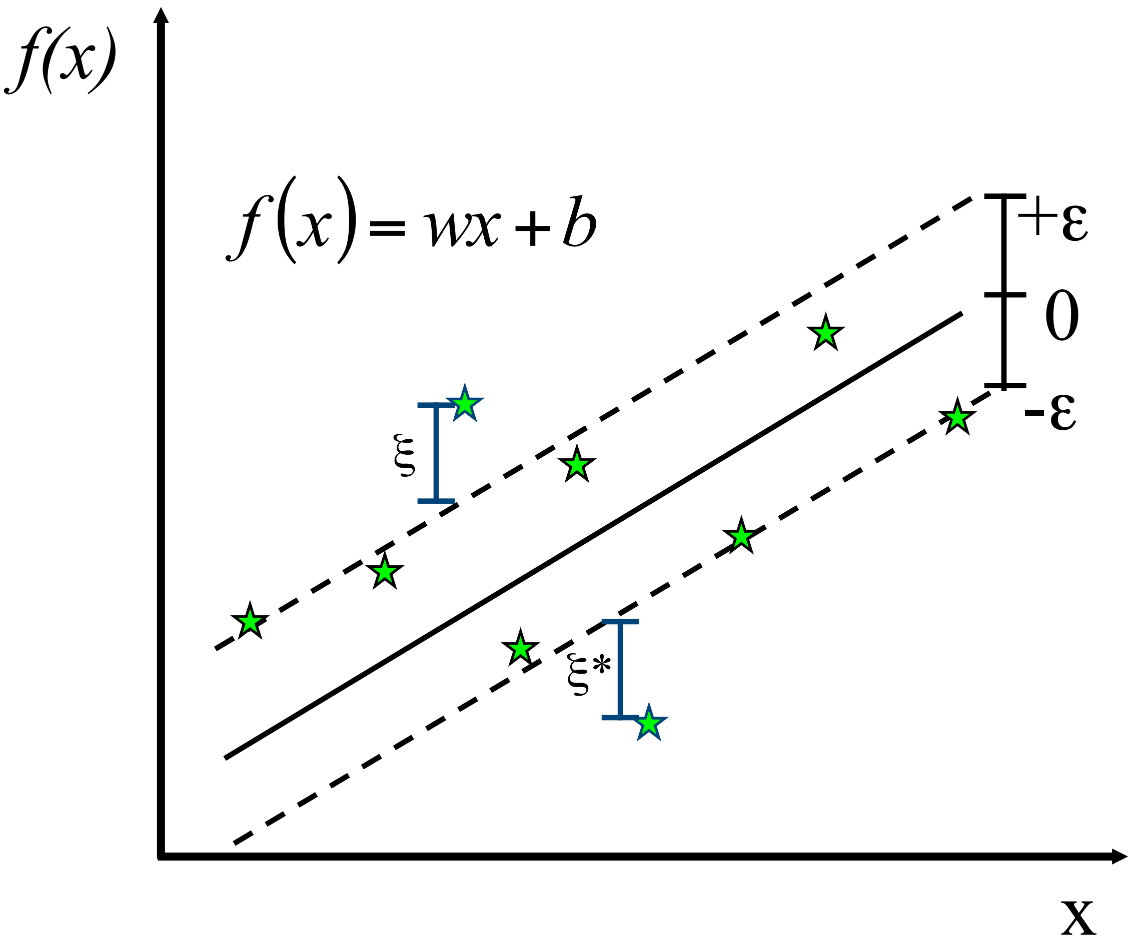
$$Min \frac{1}{2} w^T w$$

Constraints:

$$y_i - w^T x_i - b \leq \varepsilon$$

$$w^T x_i + b - y_i \leq \varepsilon$$

Support Vector Regression (SVR)



Minimise:

$$\frac{1}{2} w^T w + \boxed{C \sum_{i=1}^{N} \left(\xi_i + \xi_i^* \right)}$$

Constraints:

$$y_i - w^T x_i - b \le \varepsilon \boxed{+ \xi_i}$$

$$w^T x_i + b - y_i \le \varepsilon \boxed{+ \xi_i^*}$$

$$\boxed{\xi_i, \xi_i^* \ge 0}$$

Support Vector Regression

$$\min_{\mathbf{w},b,\xi,\xi^*} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

$$\text{s.t. } y_i - \mathbf{w}^\top \mathbf{x}_i - b \le \epsilon + \xi_i,\ \ \xi_i \ge 0 \qquad \forall 1 \le i \le n;$$

$$\mathbf{w}^\top \mathbf{x}_i + b - y_i \le \epsilon + \xi_i^*,\ \ \xi_i^* \ge 0 \quad \forall 1 \le i \le n.$$

- y_i is not -1 or 1 anymore, now it is a value
- ε is the tollerance of our function value

From Binary to Multiclass classifiers

- Three different approaches:
- **ONE-vs-ALL (OVA)**
  - Given the example sets, {E1, E2, E3, …} for the categories: {C1, C2, C3,…} the binary classifiers: {b1, b2, b3,…} are built.
  - For b1, E1 is the set of positives and E2∪E3 ∪… is the set of negatives, and so on
  - For testing: given a classification instance x, the category is the one associated with the maximum margin among all binary classifiers

From Binary to Multiclass classifiers

- **ALL-vs-ALL (AVA)**
  - Given the examples: {E1, E2, E3, …} for the categories {C1, C2, C3,…}
    - build the binary classifiers:

      {b1_2, b1_3,…, b1_n, b2_3, b2_4,…, b2_n,…,bn-1_n}
    - by learning on E1 (positives) and E2 (negatives), on E1 (positives) and E3 (negatives) and so on…
  - For testing: given an example x,
    - all the votes of all classifiers are collected
    - where b_{E1E2} = 1 means a vote for C1 and b_{E1E2} = -1 is a vote for C2
  - Select the category that gets more votes

Natural Language Processing

Part-of-Speech tagging

- Given a sentence $W_1 \ldots W_n$ and a tagset of lexical categories, find the most likely tag $T_1..T_n$ for each word in the sentence
- Example

  Secretariat/NNP is/VBZ expected/VBN to/TO race/VB tomorrow/NN

  People/NNS continue/VBP to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN
- Note that many of the words may have unambiguous tags
  - But enough words are either ambiguous or unknown that it's a nontrivial task

Part Of Speech (POS) Tagging

- Annotate each word in a sentence with a part-of-speech.

I	ate	the	spaghetti	with	meatballs.
Pro	V	Det	N	Prep	N

John	saw	the	saw	and	decided	to	take	it	to	the	table.
PN	V	Det	N	Con	V	Part	V	Pro	Prep	Det	N

- Useful for subsequent syntactic parsing and word sense disambiguation.

PTB Tagset (36 main tags + punctuation tags)

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

Solution

- Text Classifier:
  - Tags categories
  - Features windows of words around the target word
  - N-grams

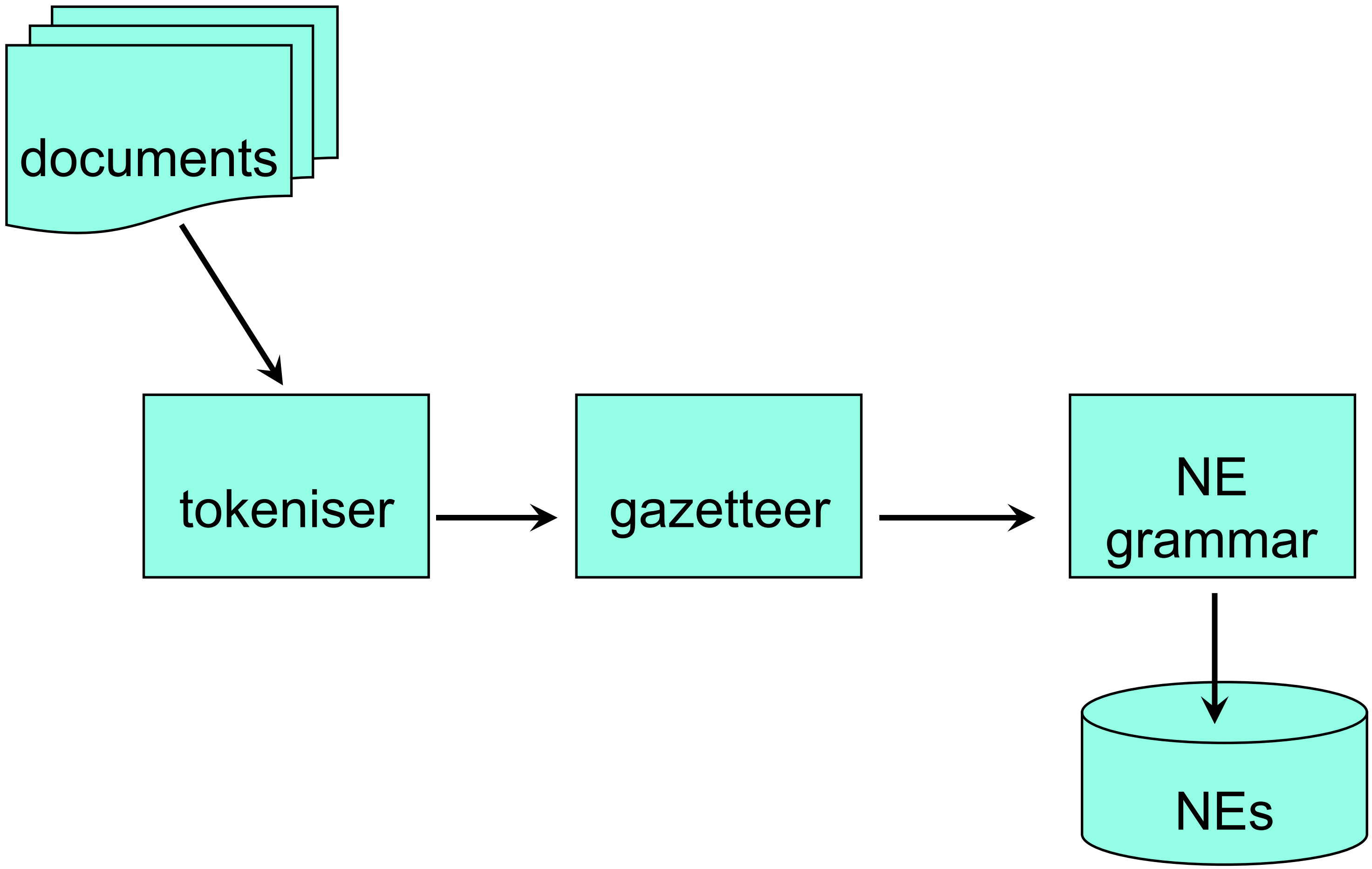
Named Entity Recognition

- NE involves **identification** of *proper names* in texts, and **classification** into a set of predefined categories of interest.
- Three universally accepted categories: **person**, **location** and **organisation**
- Other common tasks: recognition of date/time expressions, measures (percent, money, weight etc), email addresses etc.
- Other domain-specific entities: names of drugs, medical conditions, names of ships, bibliographic references etc.

Problems in NE Task Definition

- Category definitions are intuitively quite clear, but there are many grey areas.
- Many of these grey area are caused by **metonymy**.
  - Organisation vs. Location : "**England** won the World Cup" vs. "The World Cup took place in **England**".
  - Company vs. Artefact: "shares in **MTV**" vs. "watching **MTV**"
  - Location vs. Organisation: "she met him at **Heathrow**" vs. "the **Heathrow** authorities"

NE System Architecture



Approach (1)

- Again Text Categorization
- N-grams in a window centered on the NER
- Additional Features
  - Gazetteer
  - Capitalize
  - Beginning of the sentence
  - Is it all capitalized

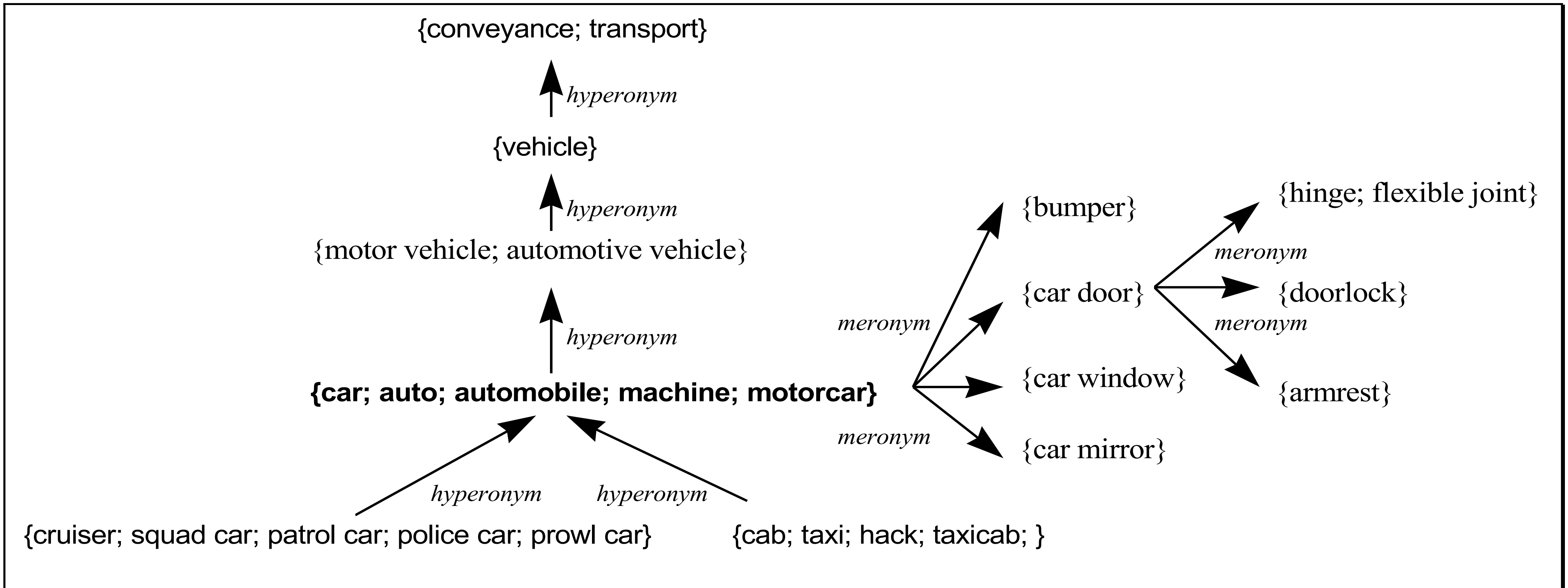
Approach (2)

- NE task in two parts:
  - Recognising the entity boundaries
  - Classifying the entities in the NE categories
- Some work is only on one task or the other
- Tokens in text are often coded with the IOB scheme
  - O – outside, B-XXX – first word in NE, I-XXX – all other words in NE
  - Easy to convert to/from inline MUC-style markup
Argentina	B-LOC
played	O
with	O
Del	B-PER
Bosque	I-PER

WordNet

- Developed at Princeton by George Miller and his team as a model of the mental lexicon.
- Semantic network in which concepts are defined in terms of relations to other concepts.
- Structure:
  - organized around the notion of synsets (sets of synonymous words)
  - basic semantic relations between these synsets
  - Initially no glosses
  - Main revision after tagging the Brown corpus with word meanings: SemCor.
  - http://www.cogsci.princeton.edu/~wn/w3wn.html

Structure

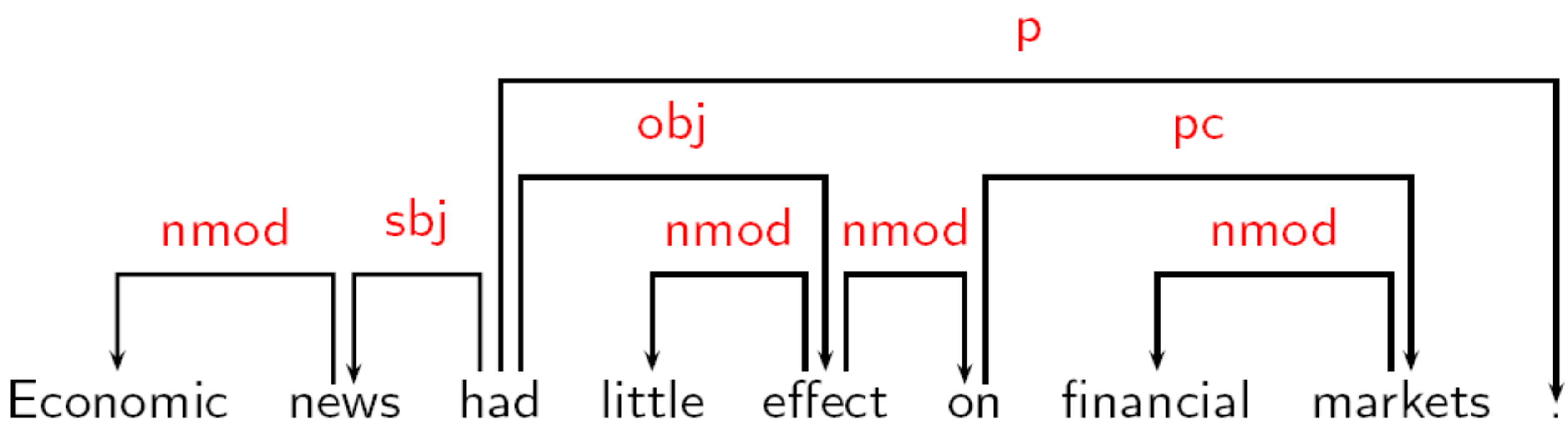

{conveyance; transport}
hyperonym
{vehicle}
hyperonym
{motor vehicle; automotive vehicle}
hyperonym
{car; auto; automobile; machine; motorcar}
hyperonym
hyperonym
{cruiser; squad car; patrol car; police car; prowl car}
{cab; taxi; hack; taxicab; }
meronym
meronym
{bumper}
{car door}
{car window}
{car mirror}
{hinge; flexible joint}
meronym
{doorlock}
meronym
{armrest}

Syntactic Parsing

Dependency Syntax

- The basic idea:
  - Syntactic structure consists of lexical items, linked by binary asymmetric relations called dependencies.

Dependency Structure

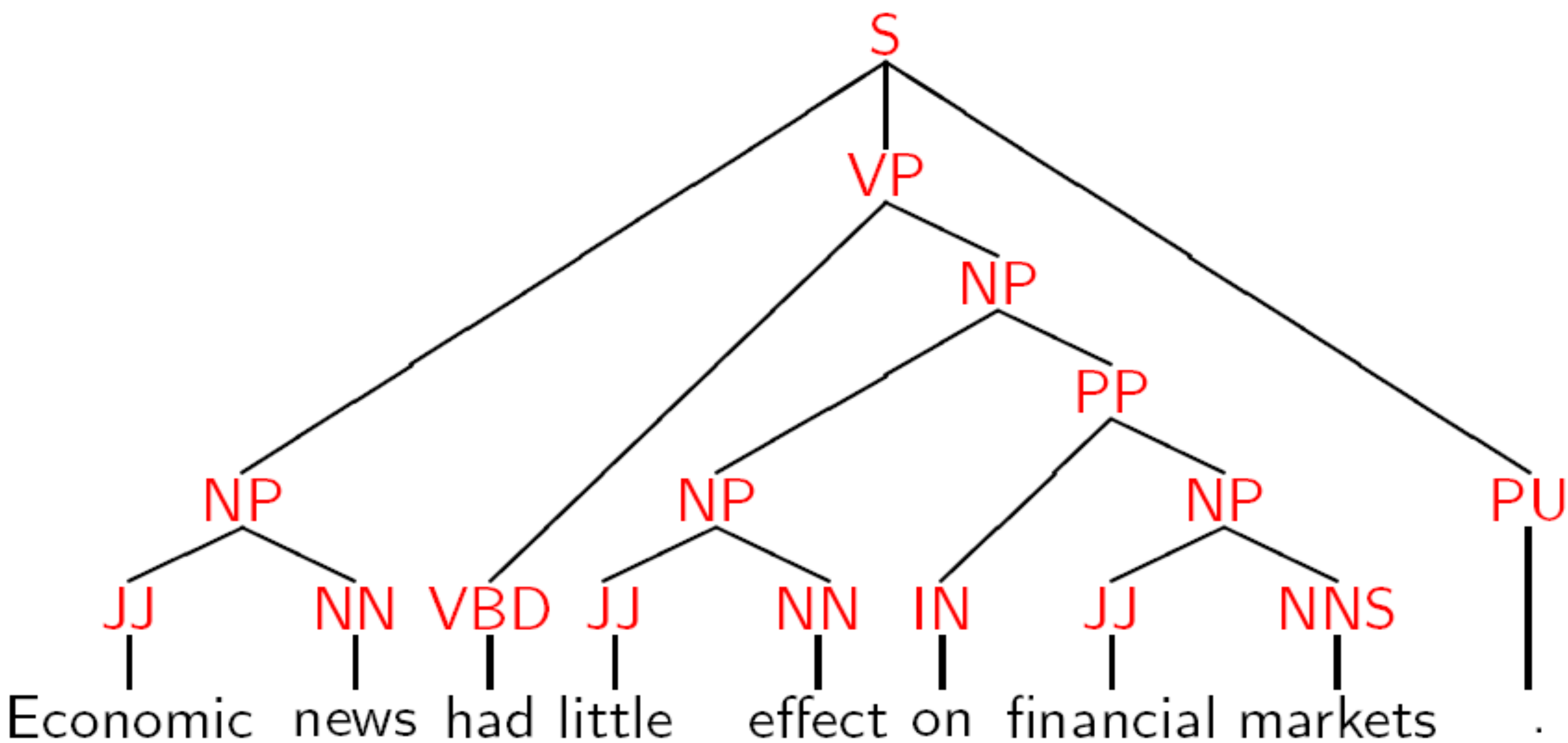


Terminology

Superior	Inferior
Head	Dependent
Governor	Modifier
Regent	Subordinate
⋮	⋮

Phrase Structure

(or Constituent Structure)



Comparison

- Dependency structures explicitly represent
  - head-dependent relations (directed arcs),
  - functional categories (arc labels),
  - possibly some structural categories (parts-of-speech).
- Phrase structures explicitly represent
  - phrases (nonterminal nodes),
  - structural categories (nonterminal labels),
  - possibly some functional categories (grammatical functions).
- Hybrid representations may combine all elements.

Predicate Argument Structures

Shallow semantics from predicate argument structures

- In an event:
  - target words describe relation among different entities
  - the participants are often seen as predicate's arguments.
- Example:

  a phosphor gives off electromagnetic energy in this form

Shallow semantics from predicate argument structures

- In an event:
  - target words describe relation among different entities
  - the participants are often seen as predicate's arguments.
- Example:

  [$_{Arg0}$ a phosphor] [$_{predicate}$ gives off] [$_{Arg1}$ electromagnetic energy] [$_{ArgM}$ in this form]

Shallow semantics from predicate argument structures

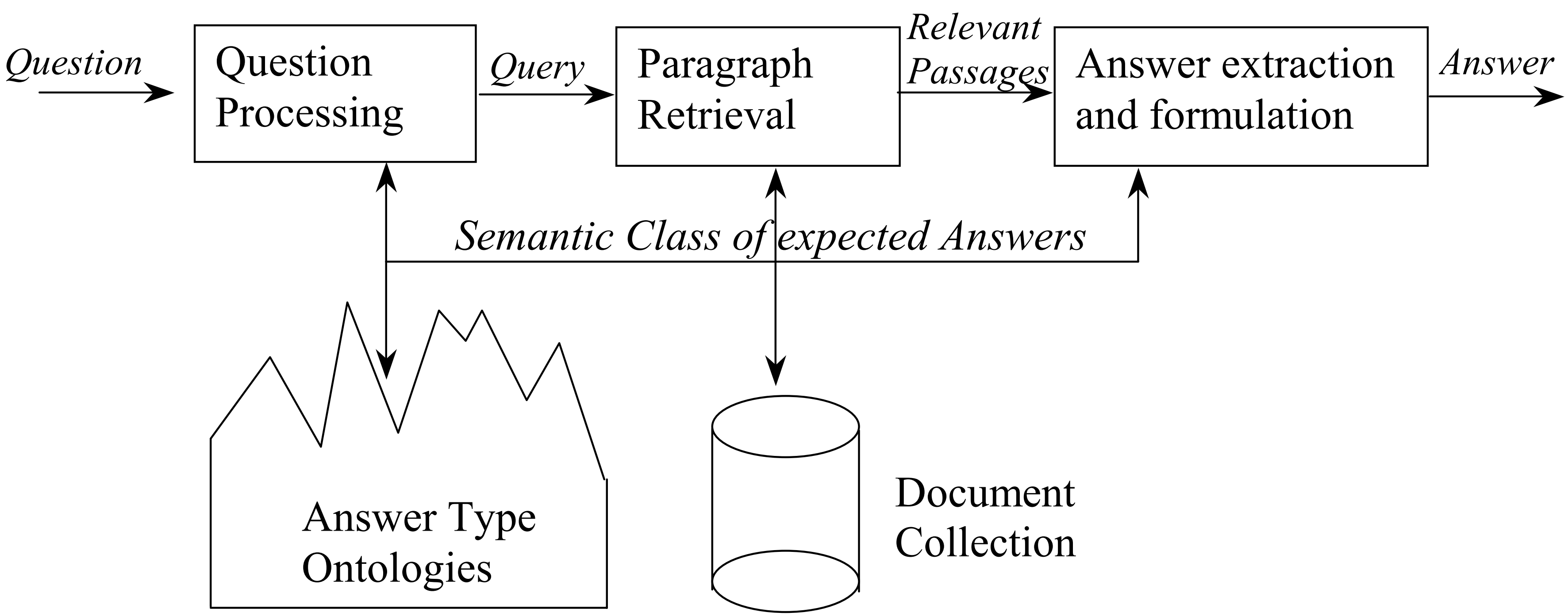
- In an event:
  - target words describe relation among different entities
  - the participants are often seen as predicate's arguments.
- Example:

  [$_{Arg0}$ a phosphor] [$_{predicate}$ gives off] [$_{Arg1}$ electromagnetic energy] [$_{ArgM}$ in this form]

  [$_{ARGM}$ When] [$_{predicate}$ hit] [$_{Arg0}$ by electrons] [$_{Arg1}$ a phosphor]

Question Answering

Basic Pipeline



Question Classification

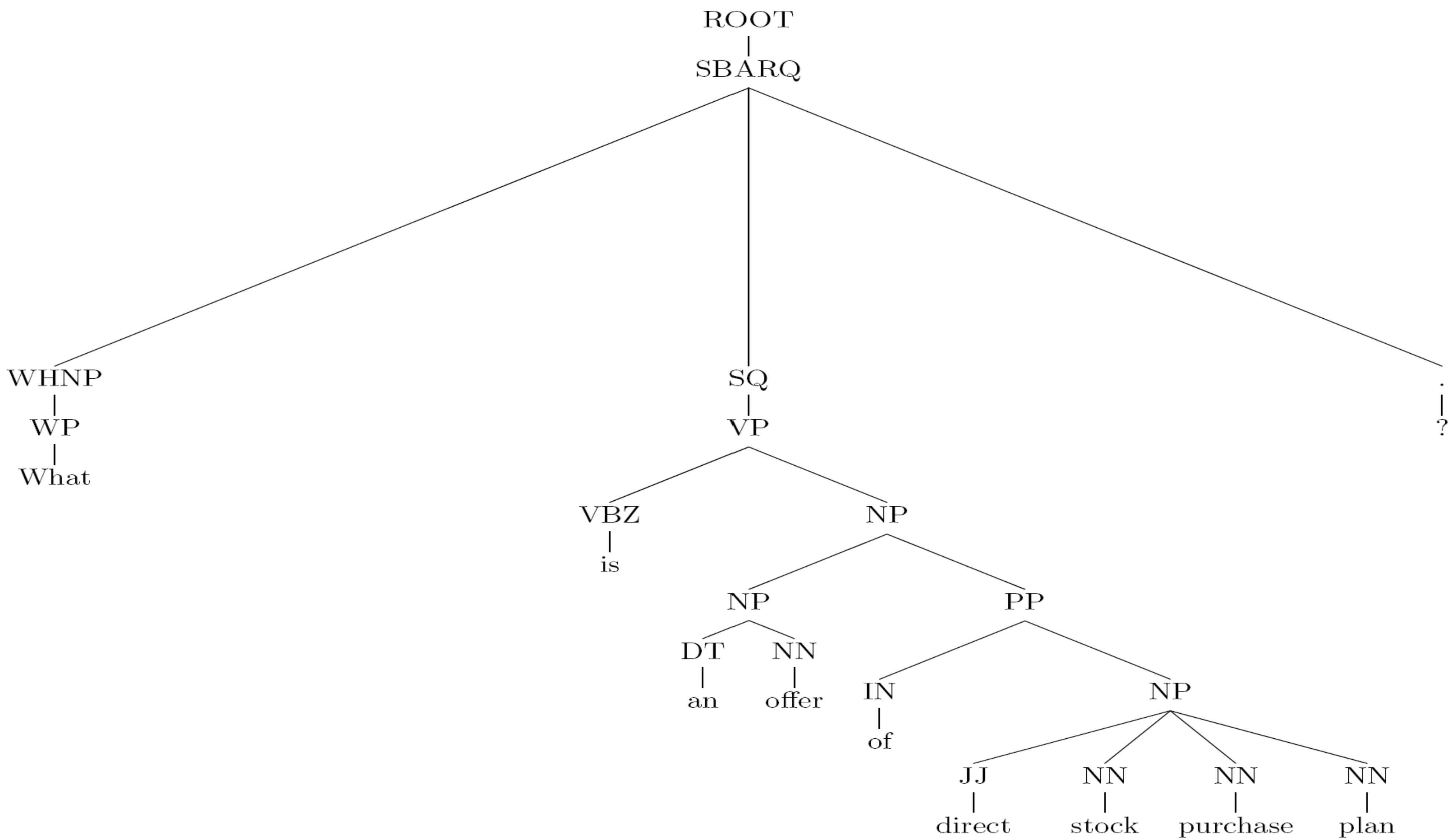
- **Definition**: What does HTML stand for?
- **Description**: What's the final line in the Edgar Allan Poe poem "The Raven"?
- **Entity**: What foods can cause allergic reaction in people?
- **Human**: Who won the Nobel Peace Prize in 1992?
- **Location**: Where is the Statue of Liberty?
- **Manner**: How did Bob Marley die?
- **Numeric**: When was Martin Luther King Jr. born?
- **Organization**: What company makes Bentley cars?

Question Classifier based on Tree Kernels

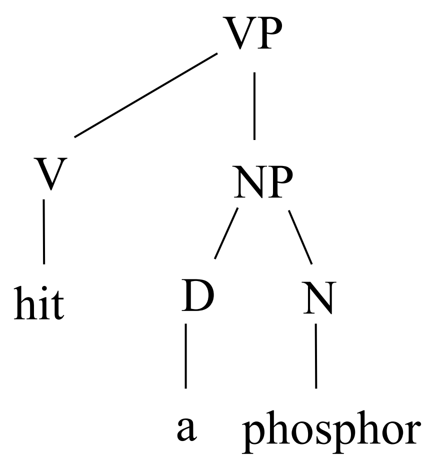
- Question dataset (http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/) [Lin and Roth, 2005])
  - Distributed on 6 categories: Abbreviations, Descriptions, Entity, Human, Location, and Numeric.
- Fixed split 5500 training and 500 test questions
- Using the whole question parse trees
  - Constituent parsing
  - Example

"What is an offer of direct stock purchase plan ?"

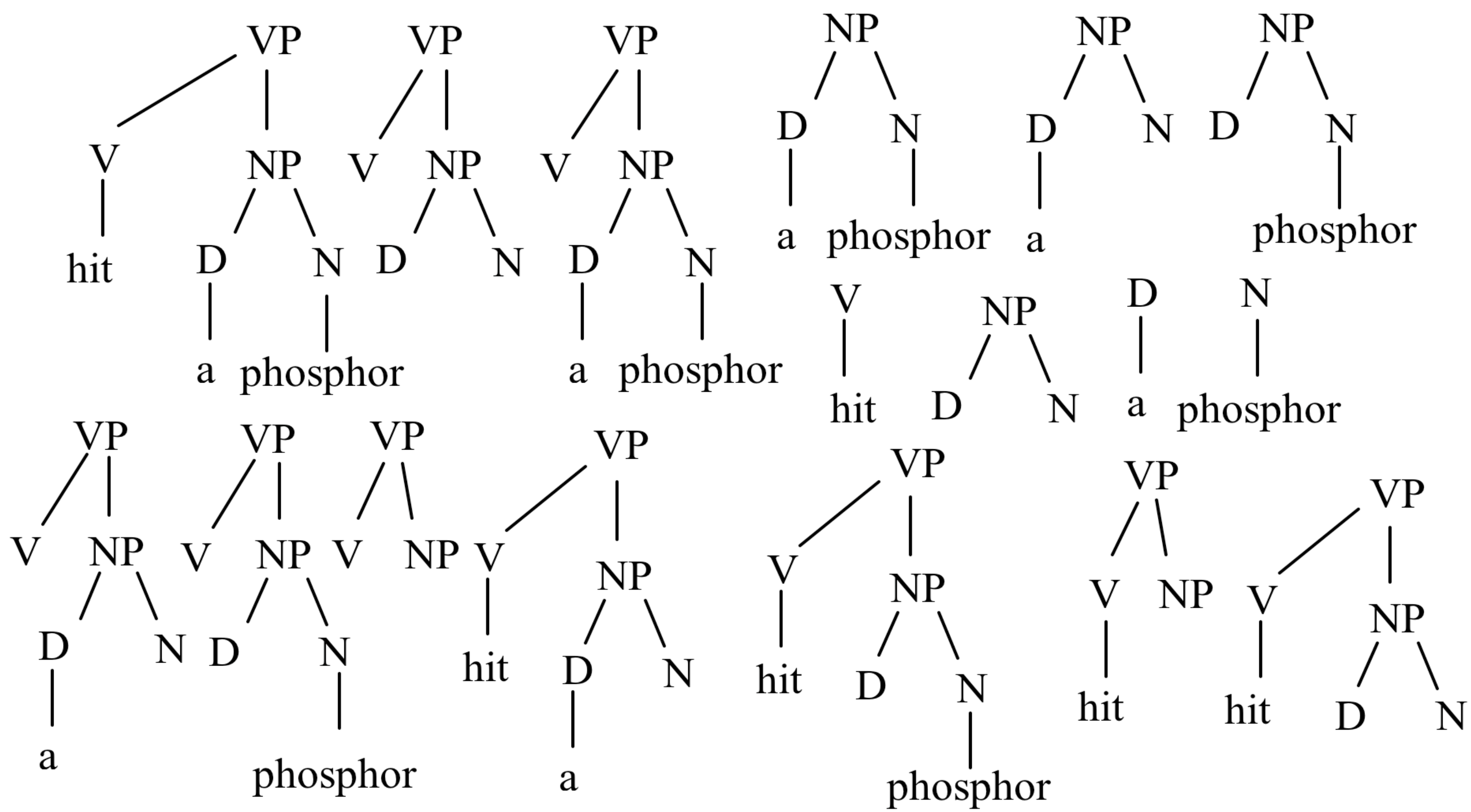
Syntactic Parse Trees (PT)



Similarity based on the number of common substructures

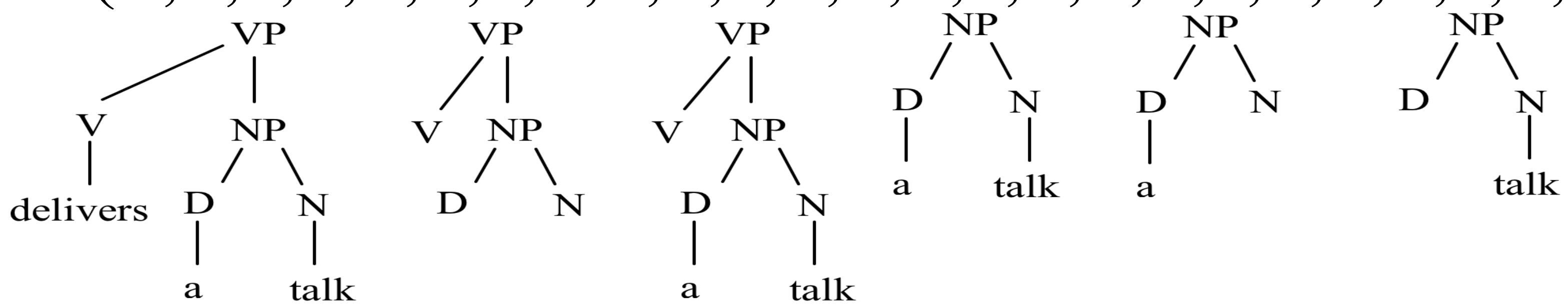


A portion of the substructure set

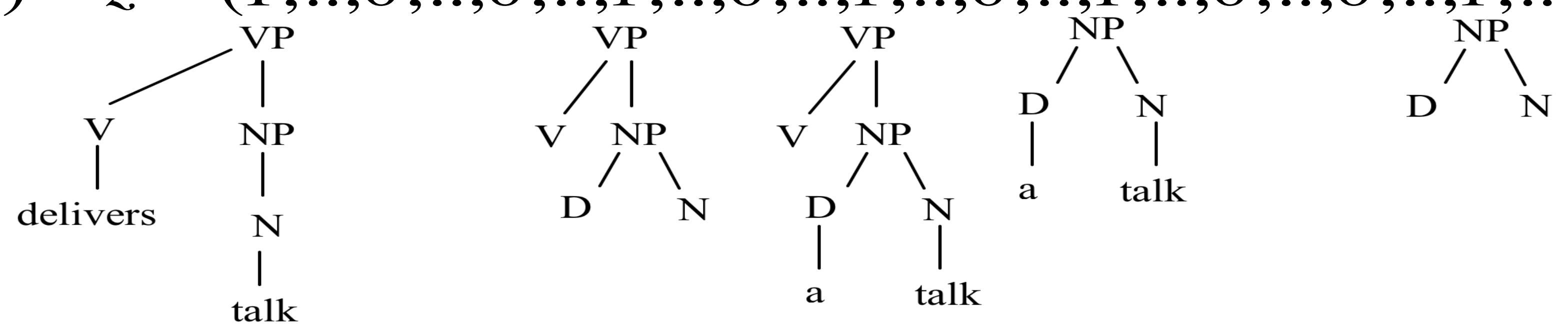


Explicit tree fragment space

$$\phi(T_x) = \vec{x} = (0,..,1,..,0,..,1,..,0,..,1,..,0,..,1,..,0,..,1,..,0,..,1,..,0)$$



$$\phi(T_z) = \vec{z} = (1,..,0,..,0,..,1,..,0,..,1,..,0,..,1,..,0,..,0,..,1,..,0,..,0)$$



- $\vec{x} \cdot \vec{z}$ counts the number of common substructures

Similarity based on WordNet

Inverted Path Length:

$$sim_{IPL}(c_1, c_2) = \frac{1}{(1 + d(c_1, c_2))^{\alpha}}$$

Wu & Palmer:

$$sim_{WUP}(c_1, c_2) = \frac{2\, dep(lso(c_1, c_2))}{d(c_1, lso(c_1, c_2)) + d(c_2, lso(c_1, c_2)) + 2\, dep(lso(c_1, c_2))}$$

Resnik:

$$sim_{RES}(c_1, c_2) = -\log\ P(lso(c_1, c_2))$$

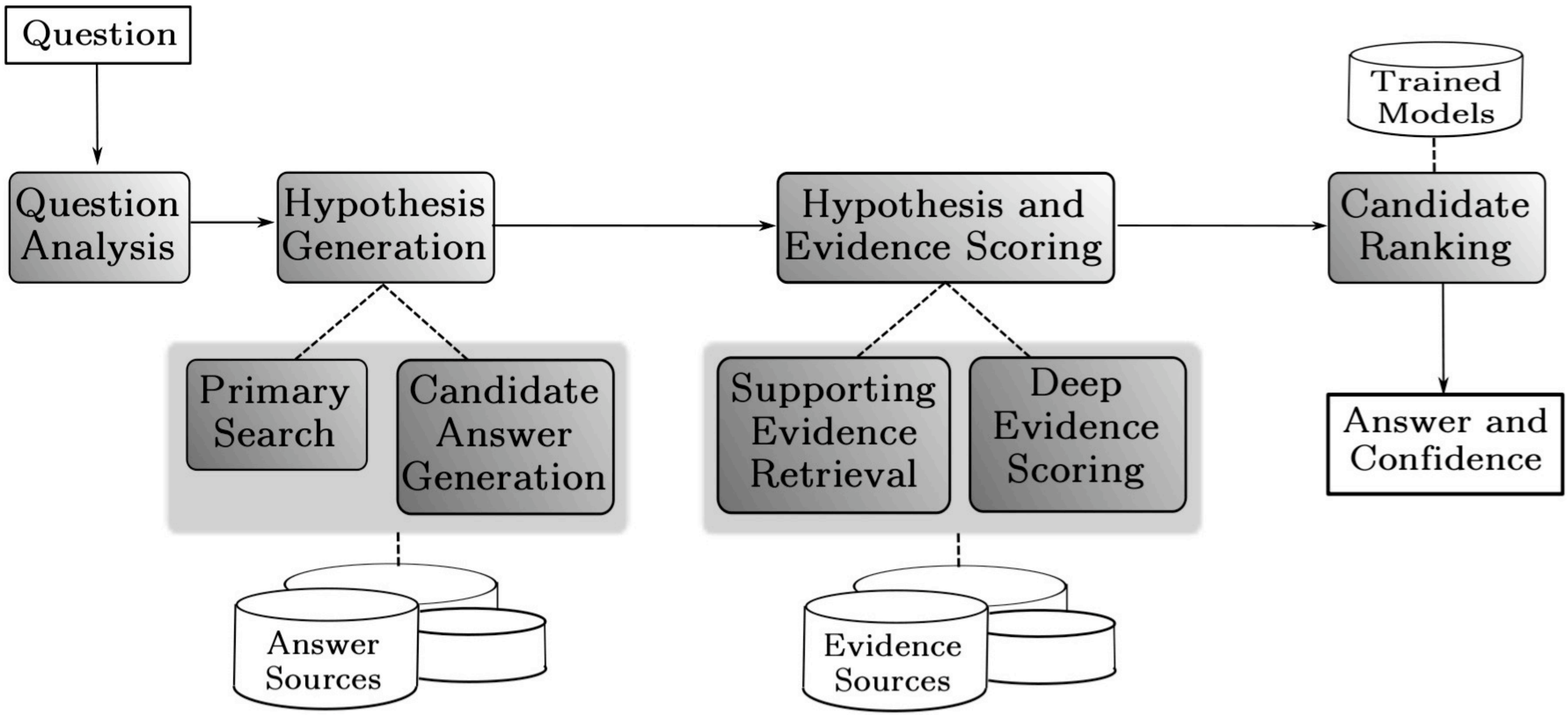
Lin:

$$sim_{LIN}(c_1, c_2) = \frac{2\ \log\ P(lso(c_1, c_2))}{\log\ P(c_1) + \log P\,(c_2)}$$

Question Classification with SSTK

	Accuracy				
λ parameter	**0.4**	**0.05**	**0.01**	**0.005**	**0.001**
linear (bow)	0.905				
string matching	0.890	0.910	**0.914**	**0.914**	0.912
full	0.904	**0.924**	0.918	0.922	0.920
full-ic	0.908	**0.922**	0.916	0.918	0.918
path-1	0.906	**0.918**	0.912	**0.918**	0.916
path-2	0.896	0.914	0.914	**0.916**	**0.916**
lin	0.908	**0.924**	0.918	0.922	0.922
wup	0.908	**0.926**	0.918	0.922	0.922

A QA Pipeline: Watson Overview



Thank you

References

- Alessandro Moschitti' handouts http://disi.unitn.eu/~moschitt/teaching.html
- Alessandro Moschitti and Silvia Quarteroni, *Linguistic Kernels for Answer Re-ranking in Question Answering Systems,* Information and Processing Management, ELSEVIER, 2010.
- Yashar Mehdad, Alessandro Moschitti and Fabio Massimo Zanzotto. *Syntactic/ Semantic Structures for Textual Entailment Recognition*. Human Language Technology - North American chapter of the Association for Computational Linguistics (HLT-NAACL), 2010, Los Angeles, Calfornia.
- Daniele Pighin and Alessandro Moschitti. *On Reverse Feature Engineering of Syntactic Tree Kernels*. In Proceedings of the 2010 Conference on Natural Language Learning, Upsala, Sweden, July 2010. Association for Computational Linguistics.
- Thi Truc Vien Nguyen, Alessandro Moschitti and Giuseppe Riccardi. *Kernel-based Reranking for Entity Extraction.* In proceedings of the 23rd International Conference on Computational Linguistics (COLING), August 2010, Beijing, China.

References

- Alessandro Moschitti. *Syntactic and semantic kernels for short text pair categorization*. In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pages 576–584, Athens, Greece, March 2009.
- Truc-Vien Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. *Convolution kernels on constituent, dependency and sequential structures for relation extraction*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1378–1387, Singapore, August 2009.
- Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. *Re-ranking models based-on small training data for spoken language understanding*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1076–1085, Singapore, August 2009.
- Alessandra Giordani and Alessandro Moschitti. *Syntactic Structural Kernels for Natural Language Interfaces to Databases*. In ECML/PKDD, pages 391–406, Bled, Slovenia, 2009.

References

- Alessandro Moschitti, Daniele Pighin and Roberto Basili*. Tree Kernels for Semantic Role Labeling,* Special Issue on Semantic Role Labeling, Computational Linguistics Journal. March 2008.
- Fabio Massimo Zanzotto, Marco Pennacchiotti and Alessandro Moschitti*, A Machine Learning Approach to Textual Entailment Recognition,* Special Issue on Textual Entailment Recognition, Natural Language Engineering, Cambridge University Press., 2008
- Mona Diab, Alessandro Moschitti, Daniele Pighin, *Semantic Role Labeling Systems for Arabic Language using Kernel Methods*. In proceedings of the 46th Conference of the Association for Computational Linguistics (ACL'08). Main Paper Section. Columbus, OH, USA, June 2008.
- Alessandro Moschitti, Silvia Quarteroni, Kernels on Linguistic Structures for Answer Extraction. In proceedings of the 46th Conference of the Association for Computational Linguistics (ACL'08). Short Paper Section. Columbus, OH, USA, June 2008.

References

- Yannick Versley, Simone Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang and Alessandro Moschitti, *BART: A Modular Toolkit for Coreference Resolution*, In Proceedings of the Conference on Language Resources and Evaluation, Marrakech, Marocco, 2008.
- Alessandro Moschitti, *Kernel Methods, Syntax and Semantics for Relational Text Categorization*. In proceeding of ACM 17th Conference on Information and Knowledge Management (CIKM). Napa Valley, California, 2008.
- Bonaventura Coppola, Alessandro Moschitti, and Giuseppe Riccardi. *Shallow semantic parsing for spoken language understanding*. In Proceedings of HLT-NAACL Short Papers, pages 85–88, Boulder, Colorado, June 2009. Association for Computational Linguistics.
- Alessandro Moschitti and Fabio Massimo Zanzotto, *Fast and Effective Kernels for Relational Learning from Texts*, Proceedings of The 24th Annual International Conference on Machine Learning (ICML 2007).

References

- Alessandro Moschitti, Silvia Quarteroni, Roberto Basili and Suresh Manandhar, *Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classification*, Proceedings of the 45th Conference of the Association for Computational Linguistics (ACL), Prague, June 2007.
- Alessandro Moschitti and Fabio Massimo Zanzotto, *Fast and Effective Kernels for Relational Learning from Texts*, Proceedings of The 24th Annual International Conference on Machine Learning (ICML 2007), Corvallis, OR, USA.
- Daniele Pighin, Alessandro Moschitti and Roberto Basili, *RTV: Tree Kernels for Thematic Role Classification*, Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-4), English Semantic Labeling, Prague, June 2007.
- Stephan Bloehdorn and Alessandro Moschitti, *Combined Syntactic and Semanitc Kernels for Text Classification*, to appear in the 29th European Conference on Information Retrieval (ECIR), April 2007, Rome, Italy.
- Fabio Aiolli, Giovanni Da San Martino, Alessandro Sperduti, and Alessandro Moschitti, *Efficient Kernel-based Learning for Trees*, to appear in the IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Honolulu, Hawaii, 2007

References

- Alessandro Moschitti, Silvia Quarteroni, Roberto Basili and Suresh Manandhar, *Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classification*, Proceedings of the 45th Conference of the Association for Computational Linguistics (ACL), Prague, June 2007.
- Alessandro Moschitti, Giuseppe Riccardi, Christian Raymond, *Spoken Language Understanding with Kernels for Syntactic/Semantic Structures*, Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU2007), Kyoto, Japan, December 2007
- Stephan Bloehdorn and Alessandro Moschitti, *Combined Syntactic and Semantic Kernels for Text Classification*, to appear in the 29th European Conference on Information Retrieval (ECIR), April 2007, Rome, Italy.
- Stephan Bloehdorn, Alessandro Moschitti: Structure and semantics for expressive text kernels. In proceeding of ACM 16th Conference on Information and Knowledge Management (CIKM-short paper) 2007: 861-864, Portugal.

References

- Fabio Aiolli, Giovanni Da San Martino, Alessandro Sperduti, and Alessandro Moschitti, *Efficient Kernel-based Learning for Trees*, to appear in the IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Honolulu, Hawaii, 2007.
- Alessandro Moschitti, *Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees.* In Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany, 2006.
- Fabio Aiolli, Giovanni Da San Martino, Alessandro Sperduti, and Alessandro Moschitti, *Fast On-line Kernel Learning for Trees*, International Conference on Data Mining (ICDM) 2006 (short paper).
- Stephan Bloehdorn, Roberto Basili, Marco Cammisa, Alessandro Moschitti, *Semantic Kernels for Text Classification based on Topological Measures of Feature Similarity*. In Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 06), Hong Kong, 18-22 December 2006. (short paper).

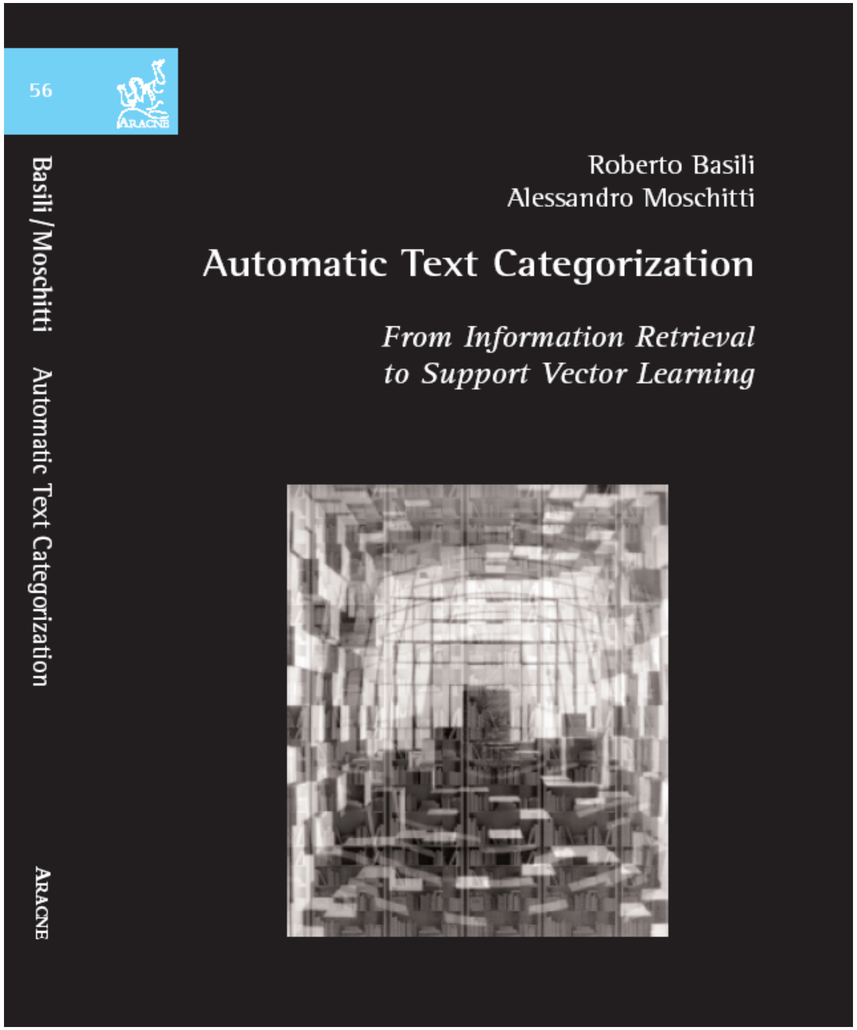
References

- Roberto Basili, Marco Cammisa and Alessandro Moschitti, *A Semantic Kernel to classify texts with very few training examples*, in Informatica, an international journal of Computing and Informatics, 2006.
- Fabio Massimo Zanzotto and Alessandro Moschitti, *Automatic learning of textual entailments with cross-pair similarities*. In Proceedings of COLING-ACL, Sydney, Australia, 2006.
- Ana-Maria Giuglea and Alessandro Moschitti*, Semantic Role Labeling via FrameNet, VerbNet and PropBank.* In Proceedings of COLING-ACL, Sydney, Australia, 2006.
- Alessandro Moschitti*, Making tree kernels practical for natural language learning.* In Proceedings of the Eleventh International Conference on European Association for Computational Linguistics, Trento, Italy, 2006.
- Alessandro Moschitti, Daniele Pighin and Roberto Basili. *Semantic Role Labeling via Tree Kernel joint inference*. In Proceedings of the 10th Conference on Computational Natural Language Learning, New York, USA, 2006.

References

- Roberto Basili, Marco Cammisa and Alessandro Moschitti, *Effective use of Wordnet semantics via kernel-based learning*. In Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005), Ann Arbor (MI), USA, 2005
- Alessandro Moschitti, *A study on Convolution Kernel for Shallow Semantic Parsing*. In proceedings of the 42-th Conference on Association for Computational Linguistic (ACL-2004), Barcelona, Spain, 2004.
- Alessandro Moschitti and Cosmin Adrian Bejan, *A Semantic Kernel for Predicate Argument Classification.* In proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004), Boston, MA, USA, 2004.

An introductory book on SVMs, Kernel methods and Text Categorization



Non-exhaustive reference list from other authors

- V. Vapnik. The Nature of Statistical Learning Theory. Springer, 1995.
- P. Bartlett and J. Shawe-Taylor, 1998. Advances in Kernel Methods - Support Vector Learning, chapter Generalization Performance of Support Vector Machines and other Pattern Classifiers. MIT Press.
- David Haussler. 1999. Convolution kernels on discrete structures. Technical report, Dept. of Computer Science, University of California at Santa Cruz.
- Lodhi, Huma, Craig Saunders, John Shawe Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. JMLR,2000
- Schölkopf, Bernhard and Alexander J. Smola. 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA.

Non-exhaustive reference list from other authors

- N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines (and other kernel-based learning methods)* Cambridge University Press, 2002
- M. Collins and N. Duffy, New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In ACL02, 2002.
- Hisashi Kashima and Teruo Koyanagi. 2002. Kernels for semi-structured data. In Proceedings of ICML'02.
- S.V.N. Vishwanathan and A.J. Smola. Fast kernels on strings and trees. In Proceedings of NIPS, 2002.
- Nicola Cancedda, Eric Gaussier, Cyril Goutte, and Jean Michel Renders. 2003. Word sequence kernels. Journal of Machine Learning Research, 3:1059–1082. D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. JMLR, 3:1083–1106, 2003.

Non-exhaustive reference list from other authors

- Taku Kudo and Yuji Matsumoto. 2003. Fast methods for kernel-based text analysis. In Proceedings of ACL'03.
- Dell Zhang and Wee Sun Lee. 2003. Question classification using support vector machines. In Proceedings of SIGIR'03, pages 26–32.
- Libin Shen, Anoop Sarkar, and Aravind k. Joshi. Using LTAG Based Features in Parse Reranking. In Proceedings of EMNLP'03, 2003
- C. Cumby and D. Roth. Kernel Methods for Relational Learning. In Proceedings of ICML 2003, pages 107–114, Washington, DC, USA, 2003.
- J. Shawe-Taylor and N. Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, 2004.
- A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In Proceedings of the 42nd Annual Meeting on ACL, Barcelona, Spain, 2004.

Non-exhaustive reference list from other authors

- Kristina Toutanova, Penka Markova, and Christopher Manning. The Leaf Path Projection View of Parse Trees: Exploring String Kernels for HPSG Parse Selection. In Proceedings of EMNLP 2004.
- Jun Suzuki and Hideki Isozaki. 2005. Sequence and Tree Kernels with Statistical Feature Mining. In Proceedings of NIPS'05.
- Taku Kudo, Jun Suzuki, and Hideki Isozaki. 2005. Boosting based parse reranking with subtree features. In Proceedings of ACL'05.
- R. C. Bunescu and R. J. Mooney. Subsequence kernels for relation extraction. In Proceedings of NIPS, 2005.
- R. C. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In Proceedings of EMNLP, pages 724–731, 2005.
- S. Zhao and R. Grishman. Extracting relations with integrated information using kernel methods. In Proceedings of the 43rd Meeting of the ACL, pages 419–426, Ann Arbor, Michigan, USA, 2005.

Non-exhaustive reference list from other authors

- J. Kazama and K. Torisawa. Speeding up Training with Tree Kernels for Node Relation Labeling. In Proceedings of EMNLP 2005, pages 137–144, Toronto, Canada, 2005.
- M. Zhang, J. Zhang, J. Su, , and G. Zhou. A composite kernel to extract relations between entities with both flat and structured features. In Proceedings of COLING-ACL 2006, pages 825–832, 2006.
- M. Zhang, G. Zhou, and A. Aw. Exploring syntactic structured features over parse trees for relation extraction using kernel methods. Information Processing and Management, 44(2):825–832, 2006.
- G. Zhou, M. Zhang, D. Ji, and Q. Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In Proceedings of EMNLP-CoNLL 2007, pages 728–736, 2007.

Non-exhaustive reference list from other authors

- Ivan Titov and James Henderson. Porting statistical parsers with data-defined kernels. In Proceedings of CoNLL-X, 2006
- Min Zhang, Jie Zhang, and Jian Su. 2006. Exploring Syntactic Features for Relation Extraction using a Convolution tree kernel. In Proceedings of NAACL.
- M. Wang. A re-examination of dependency path kernels for relation extraction. In Proceedings of the 3rd International Joint Conference on Natural Language Processing-IJCNLP, 2008.

Example on Predicate Argument Classification

- In an event:
  - target words describe relation among different entities
  - the participants are often seen as predicate's arguments.
- Example:

  Paul gives a talk in Rome

Example on Predicate Argument Classification

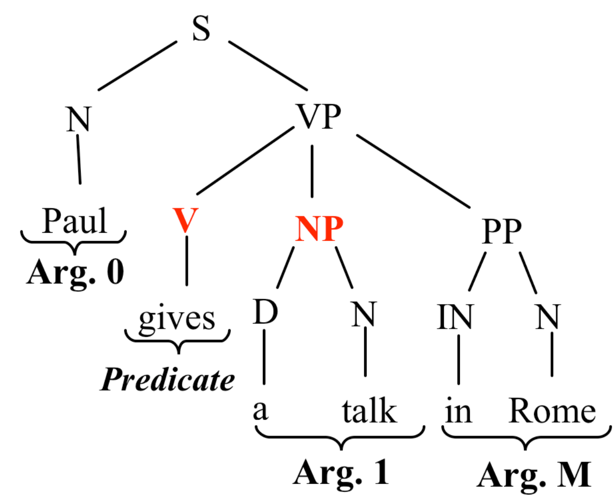
- In an event:
  - target words describe relation among different entities
  - the participants are often seen as predicate's arguments.
- Example:

  [$_{Arg0}$ Paul] [$_{predicate}$ gives] [$_{Arg1}$ a talk] [$_{ArgM}$ in Rome]

Predicate-Argument Feature Representation

Given a sentence, a predicate *p*:

1. Derive the sentence parse tree
2. For each node pair <N_p,N_x>
   a. Extract a feature representation set *F*
   b. If N_x exactly covers the Arg-i, *F* is one of its positive examples
   c. *F* is a negative example otherwise



Vector Representation for the linear kernel