Natural Language Processing: Interpretation, Reasoning and **Machine Learning**

Roberto Basili
(Università di Roma, Tor Vergata)
dblp: http://dblp.uni-trier.de/pers/hd/b/Basili:Roberto.html
Google scholar: https://scholar.google.com/citations?user=U1A22fYAAAAJ&hl=it&oi=sra

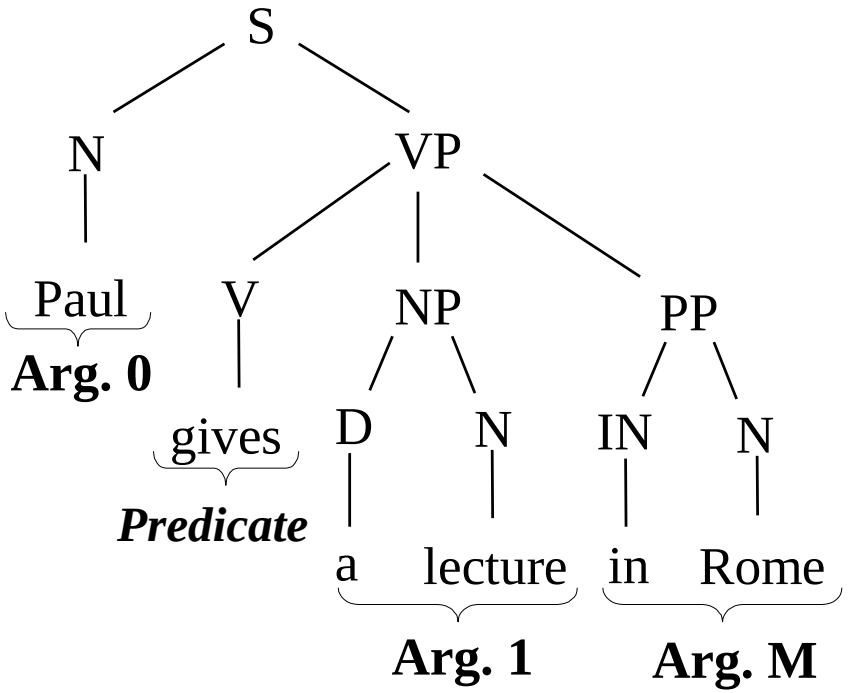
Politecnico di Milano, 4-5 Maggio 2017

Overview

- **Artificial Intelligence, Natural Language & Speech**
  - Information, Representation, (re)current challenges, success(and unsuccess)ful stories
- **Natural Language Processing: linguistic background**
- *break*
- **Natural Language Processing: Tasks, Models and Methods**
- **The Role of Machine Learning Technologies**
  - Lexicon Acquisition : Automatic Development of Dictionaries, Semantic Lexicons and Ontologies
  - Statistical Language Processing: Semantic Role Labeling
- *break*
- **Natural Language Processing : Results & Applications**
  - Semantic Document Management
  - Web-based Opinion Mining Systems, Market Watch & Brand Reputation Management
  - Human Robotic Voice Interaction

ML in NLP … a prologue

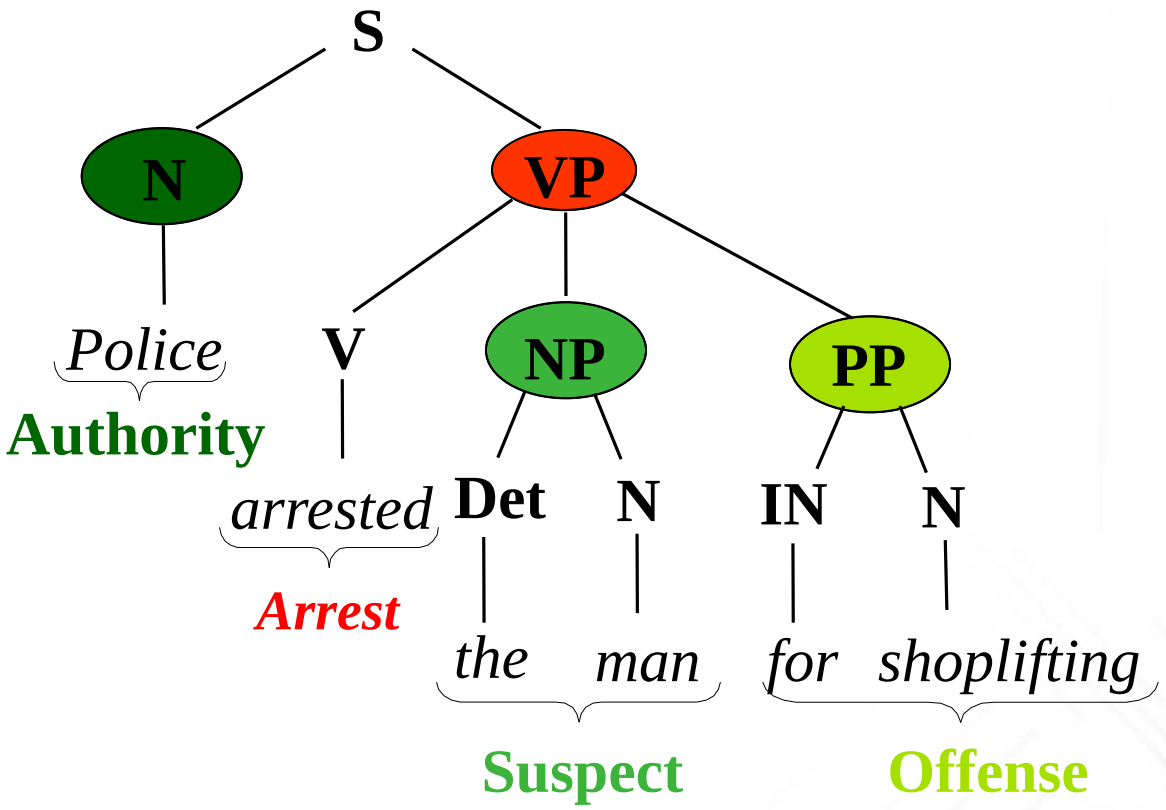
- The syntax-semantic mapping



- Different semantic theories
  (e.g. PropBank vs. FrameNet)

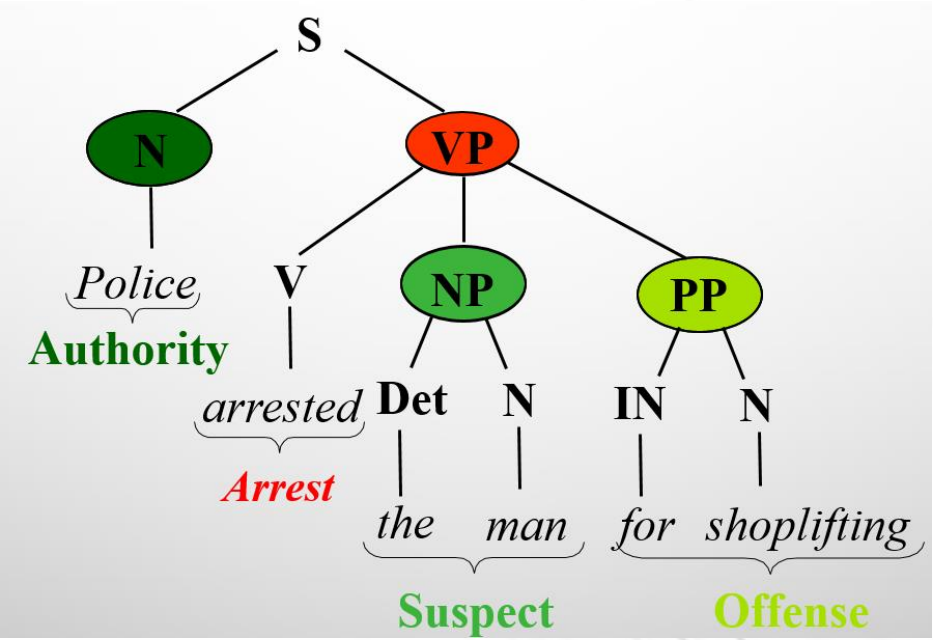
Linking syntax to semantics

- *Police arrested the man for shoplifting*

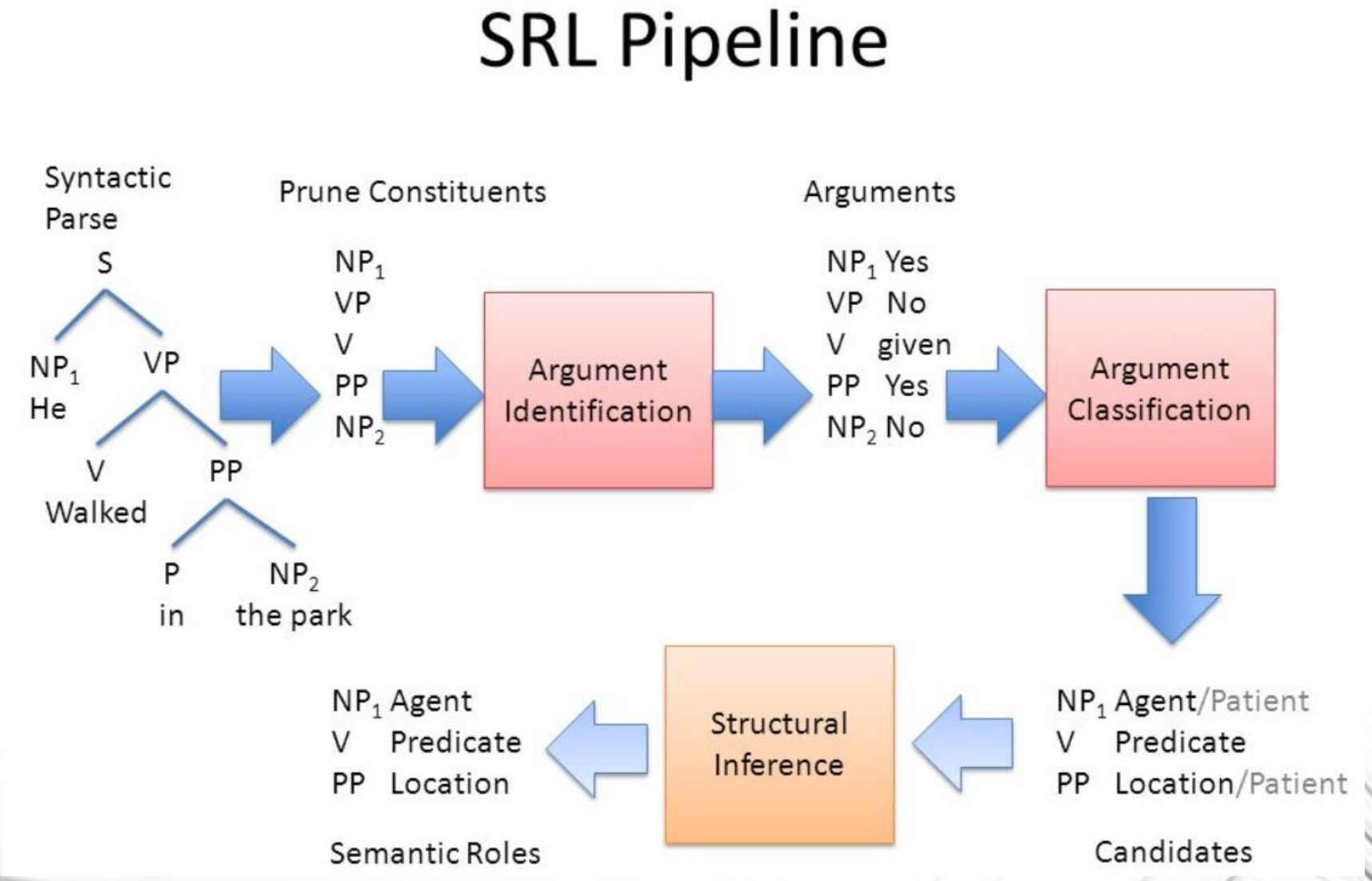


A tabular vision

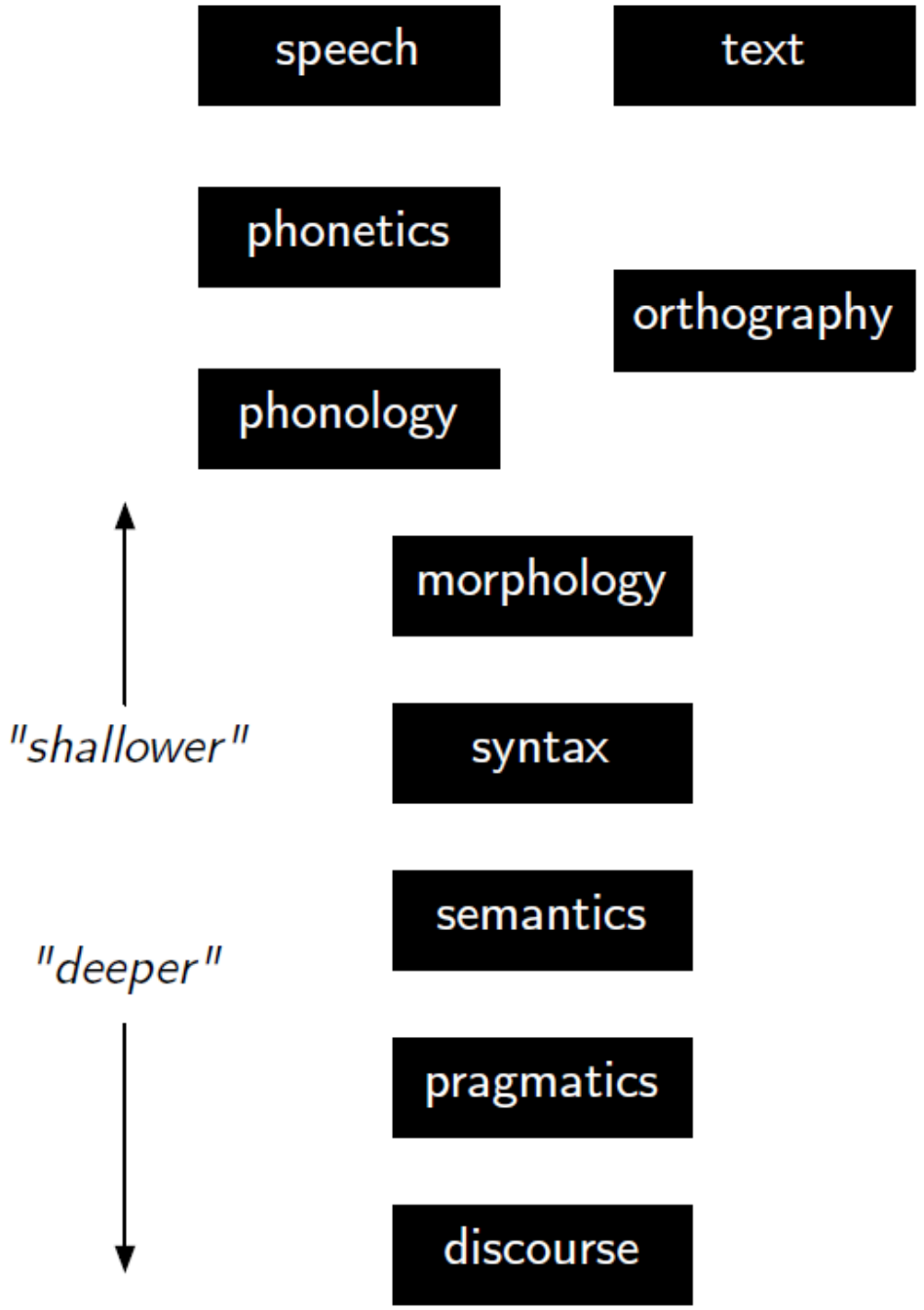
Word	Predicate	Semantic Role
Police	-	Authority
arrested	Target	Arrest
the	-	SUSPECT
man	-	SUSPECT
for	-	OFFENSE
Shoplifting	-	OFFENSE



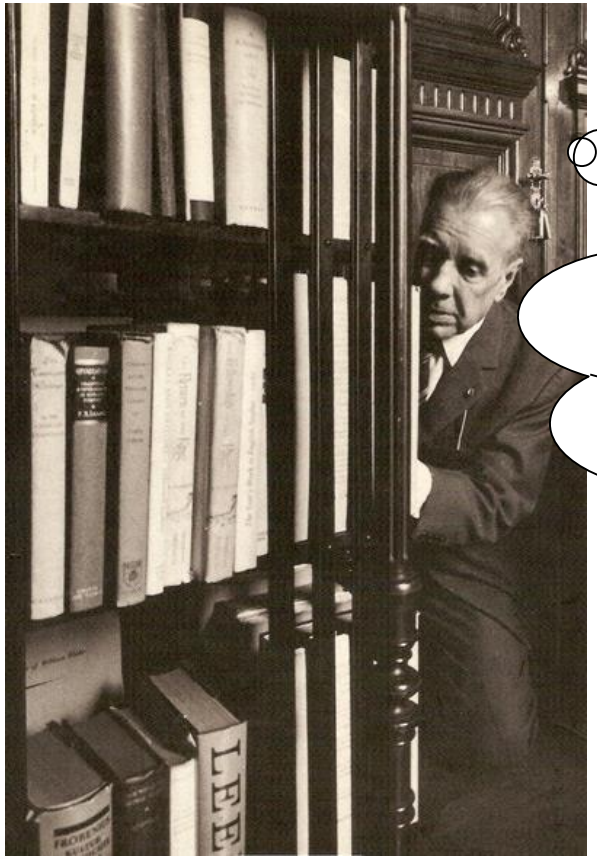
Using Framenet/PropBank



NLP: linguistic levels



Language as a system of rules



*... comincia qui la mia disperazione di scrittore. Ogni **linguaggio** è **un alfabeto di simboli** il cui uso presuppone **un passato che gli interlocutori condividono; come trasmettere agli altri l'infinito Aleph** che la mia timorosa memoria a stento abbraccia?*

(*) J.L.Borges, "*L'aleph*", 1949.

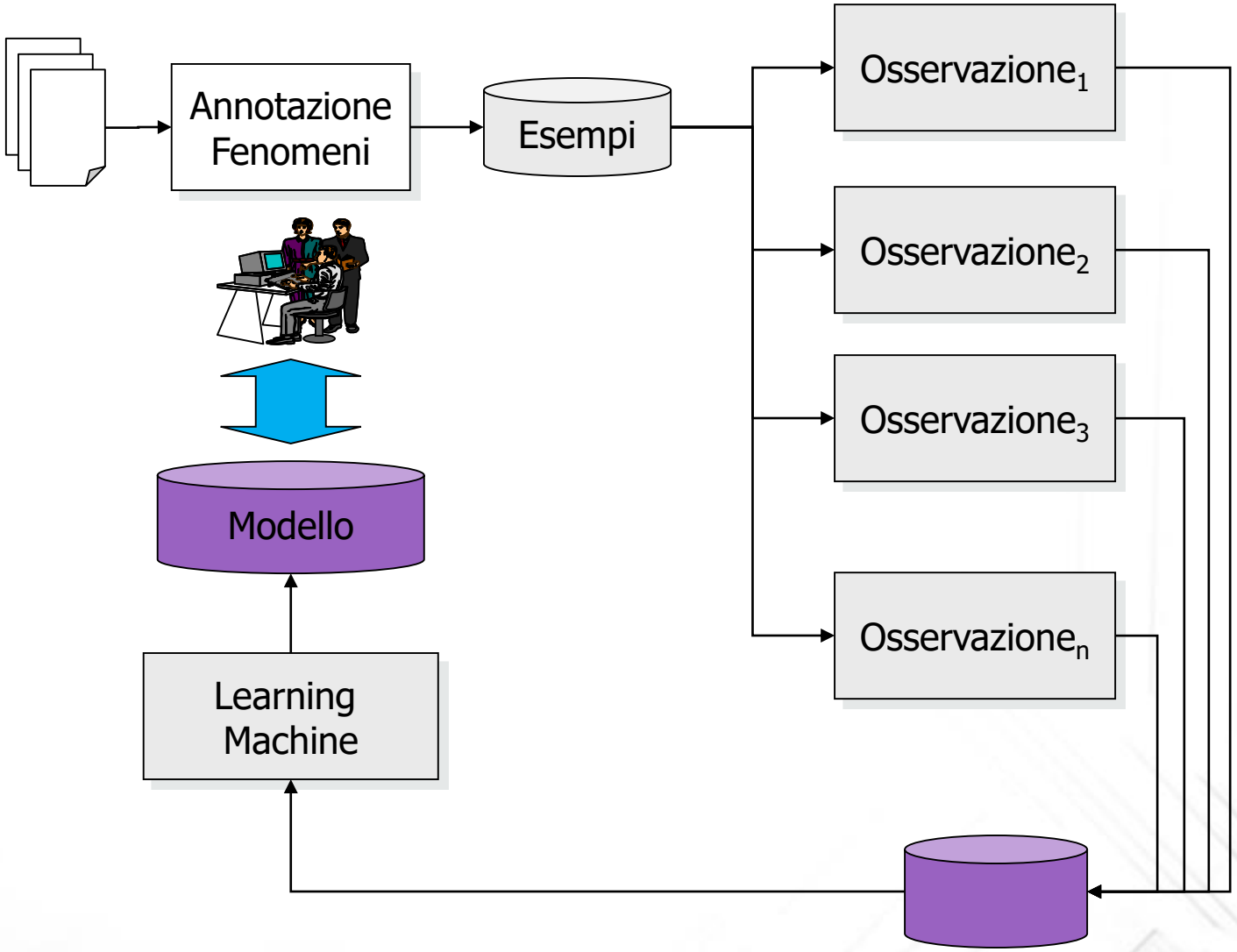
- … Meaning is acquired and recognized within the daily pactices related to its usage
  - *The meaning of a word is to be defined by the rules for its use, not by the feeling that attaches to the words*
    L. Wittgenstein's Lectures, Cambridge 1932-1935.

- Recognizing one meaning consists in the ability of mapping a linguistic expression to an experience (praxis) through mechanisms such as **analogy** or approximating **equivalence functions** or through the **minimization of the risks** of being wrong/inappropriate/obscure

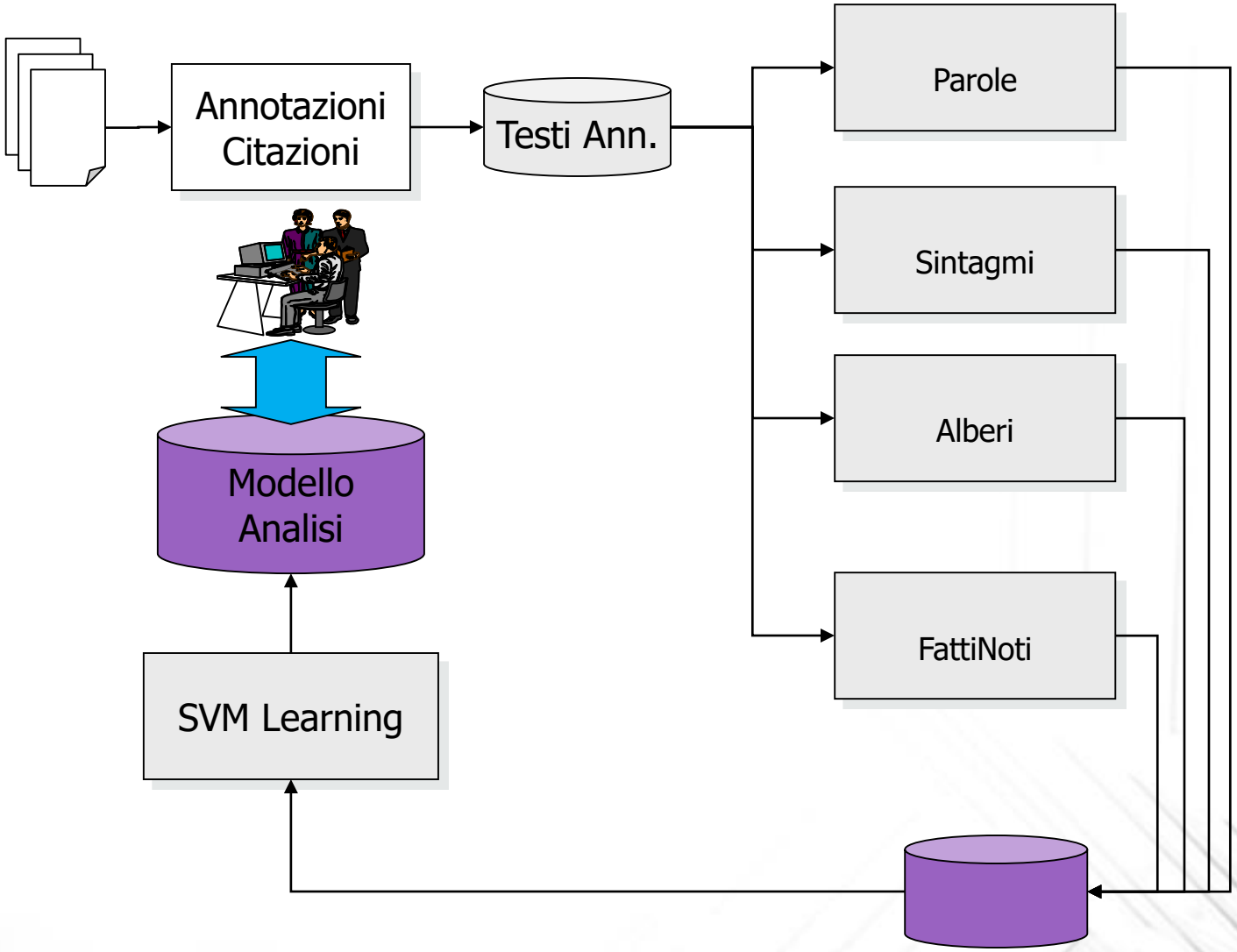
- The interpretation process can be obtained through the **induction of one (or more) decision function(s) from experience**

A different perspective

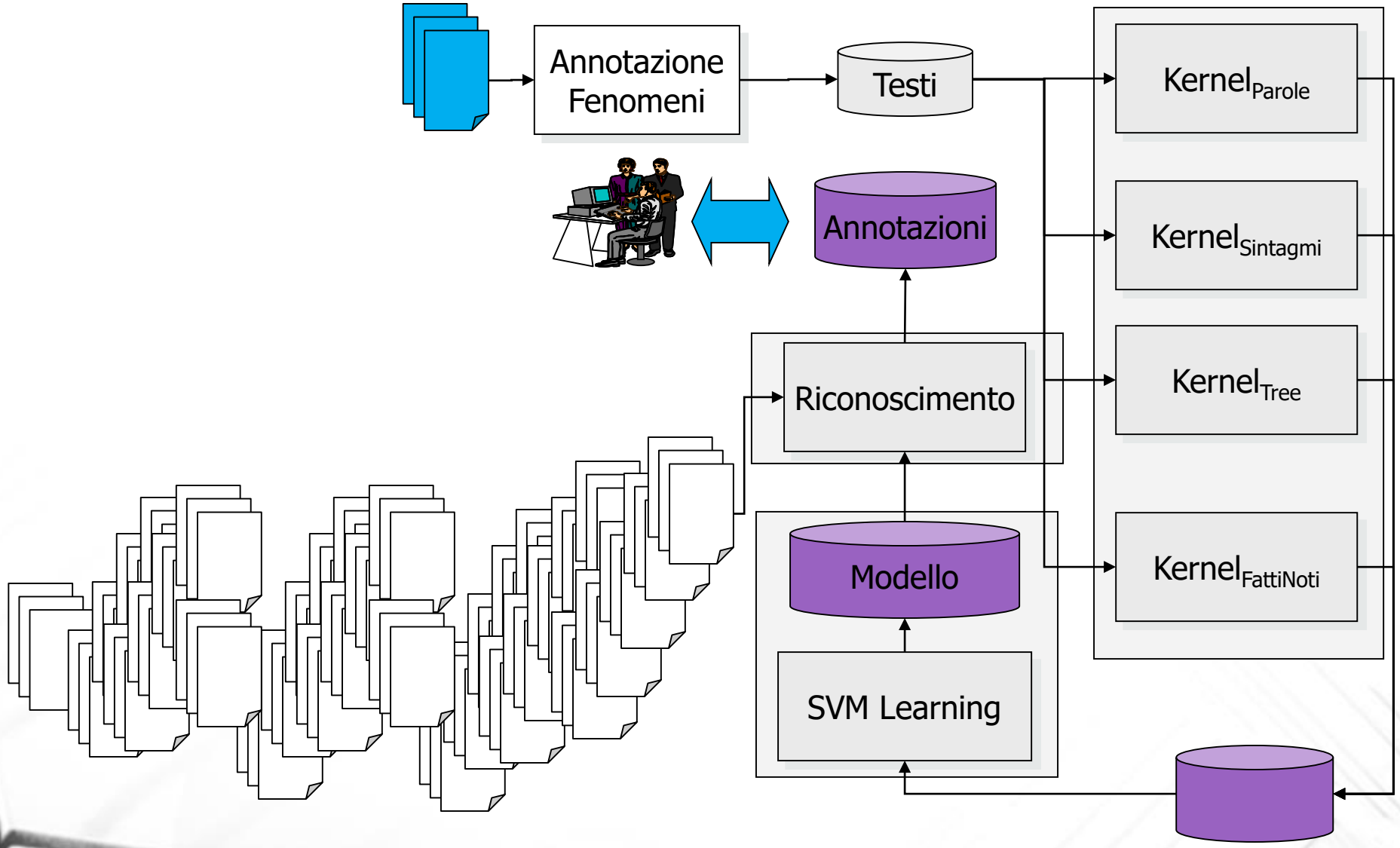
The inductive process



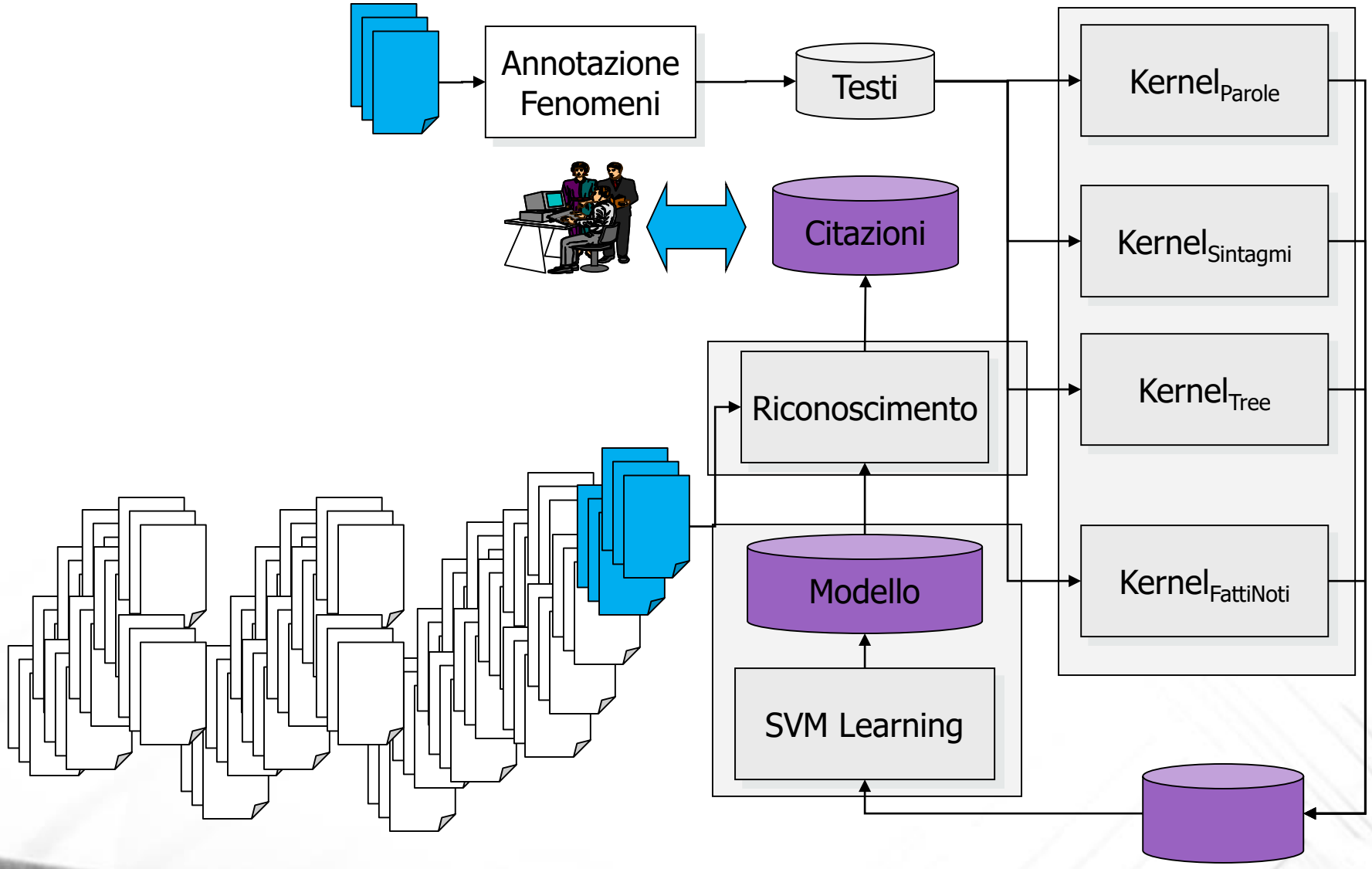
The inductive process



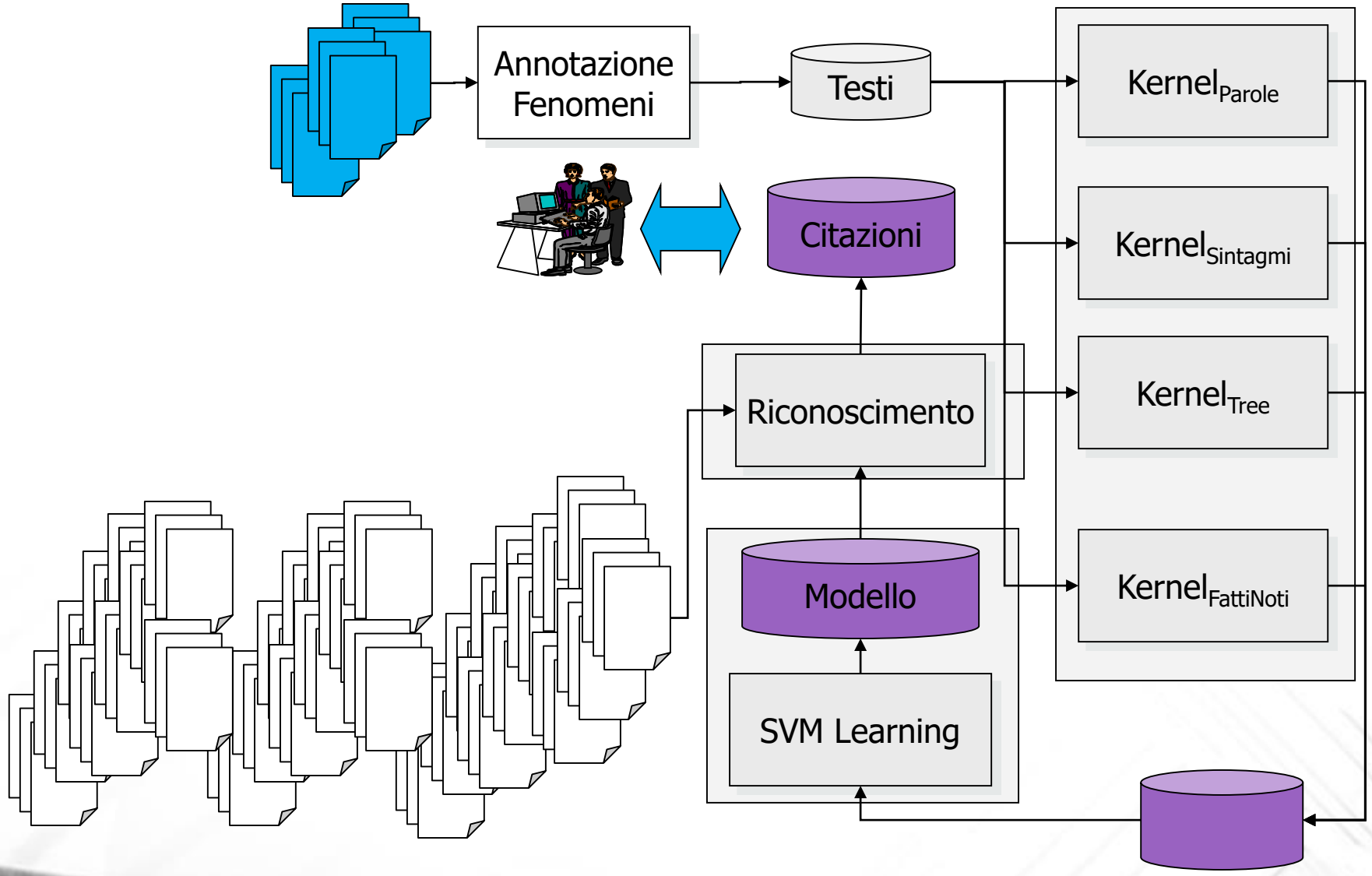
The inductive process



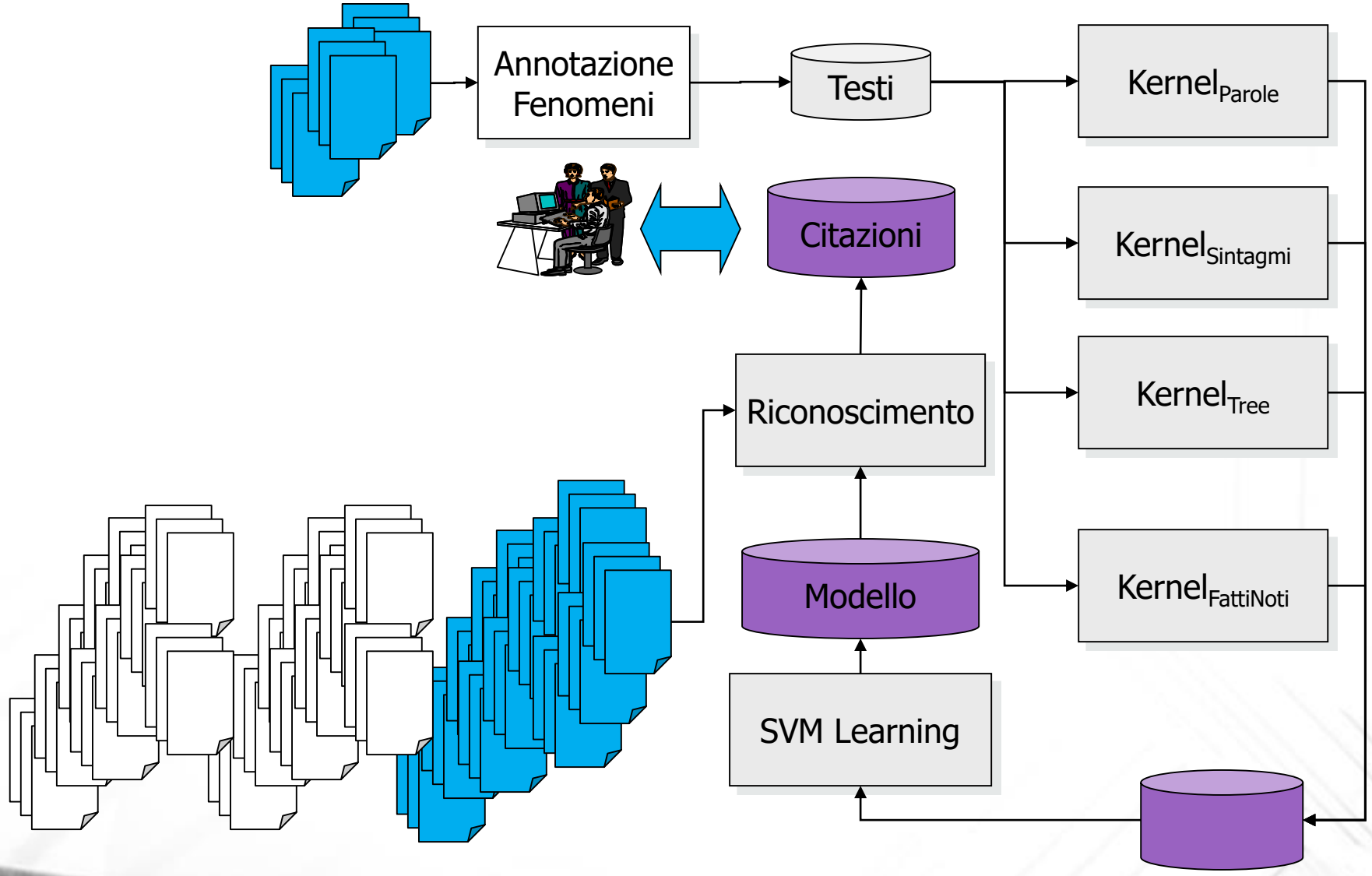
The inductive process



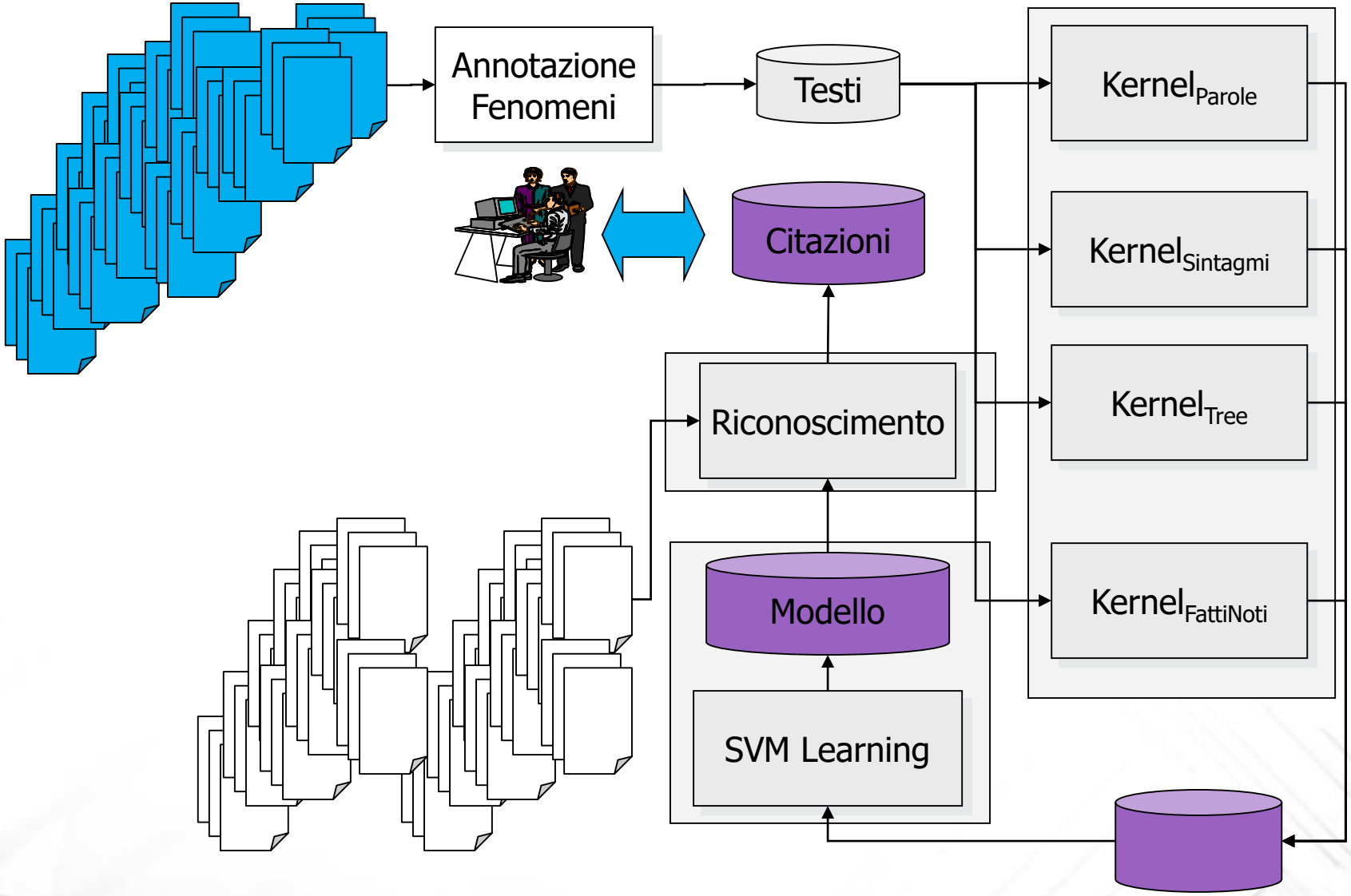
The inductive process



The inductive process



The inductive process

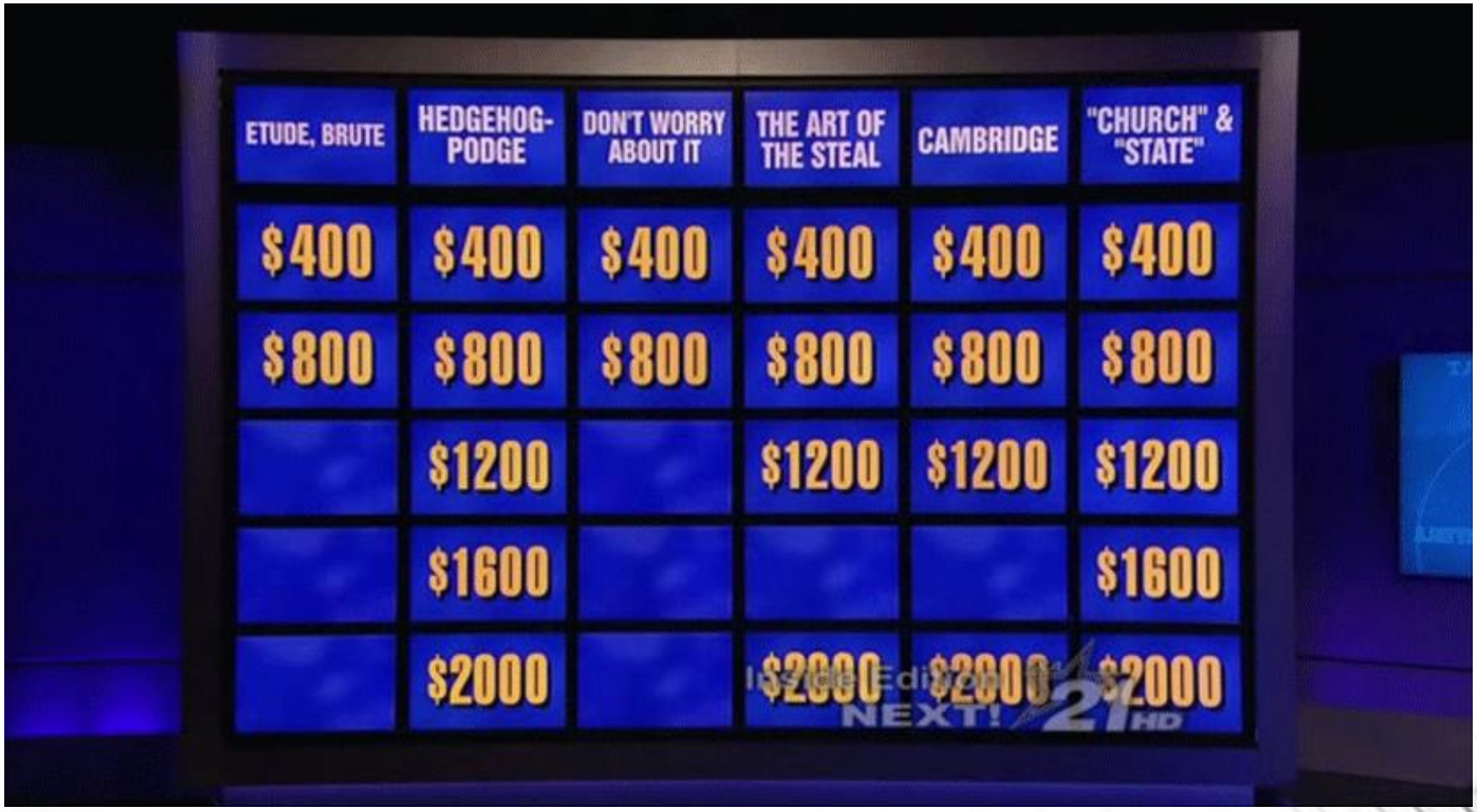


IBM Watson: between Intelligence and Data

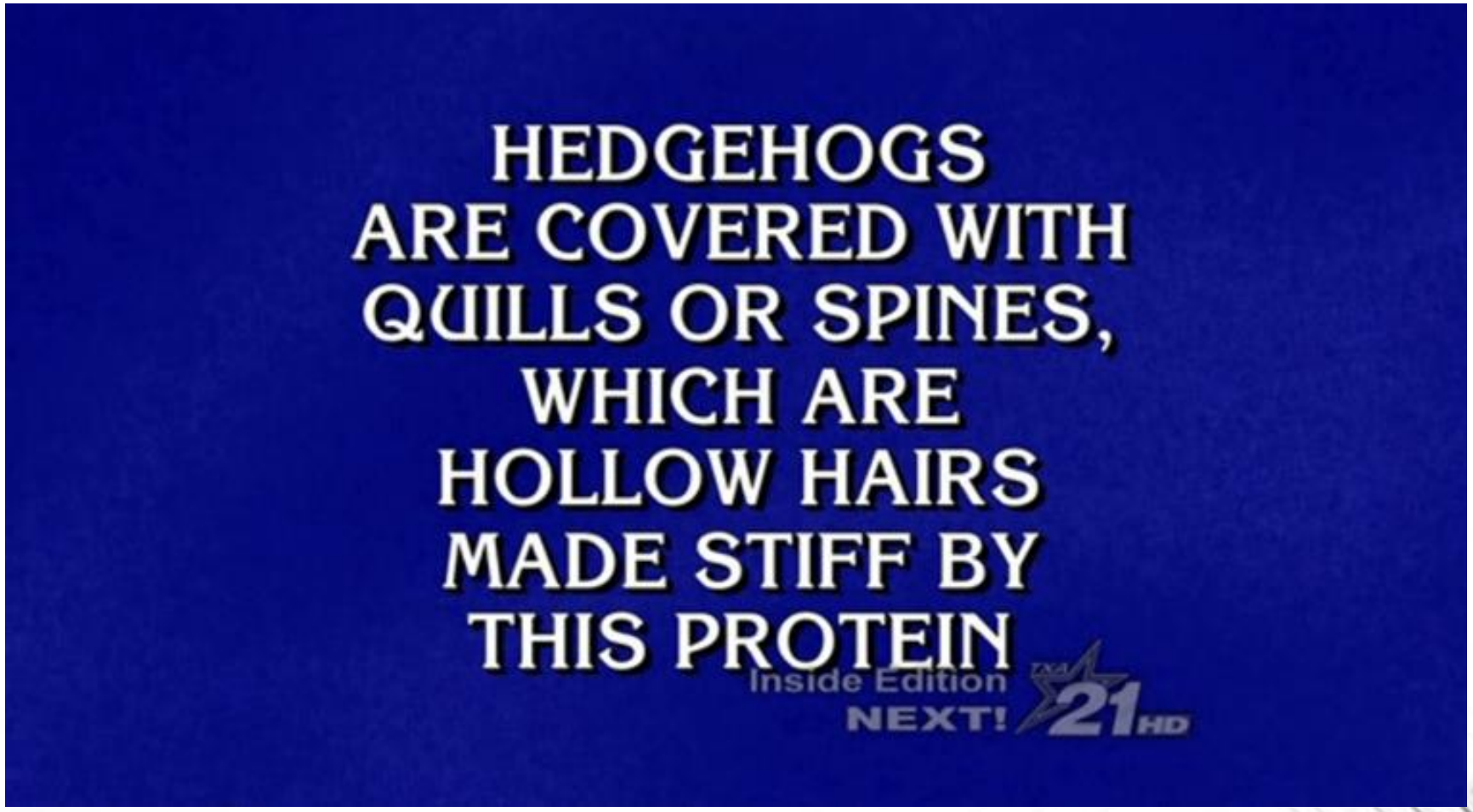
- IBM's Watson

- http://www-03.ibm.com/innovation/us/watson/science-behind_watson.shtml

Jeopardy!


ETUDE, BRUTE
HEDGEHOG-PODGE
DON'T WORRY ABOUT IT
THE ART OF THE STEAL
CAMBRIDGE
"CHURCH" & "STATE"
$400 $400 $400 $400 $400 $400
$800 $800 $800 $800 $800 $800
$1200 $1200 $1200 $1200
$1600 $1600
$2000 $2000 $2000 $2000
NEXT!
21 HD

Watson


HEDGEHOGS
ARE COVERED WITH
QUILLS OR SPINES,
WHICH ARE
HOLLOW HAIRS
MADE STIFF BY
THIS PROTEIN
Inside Edition
NEXT!
21 HD

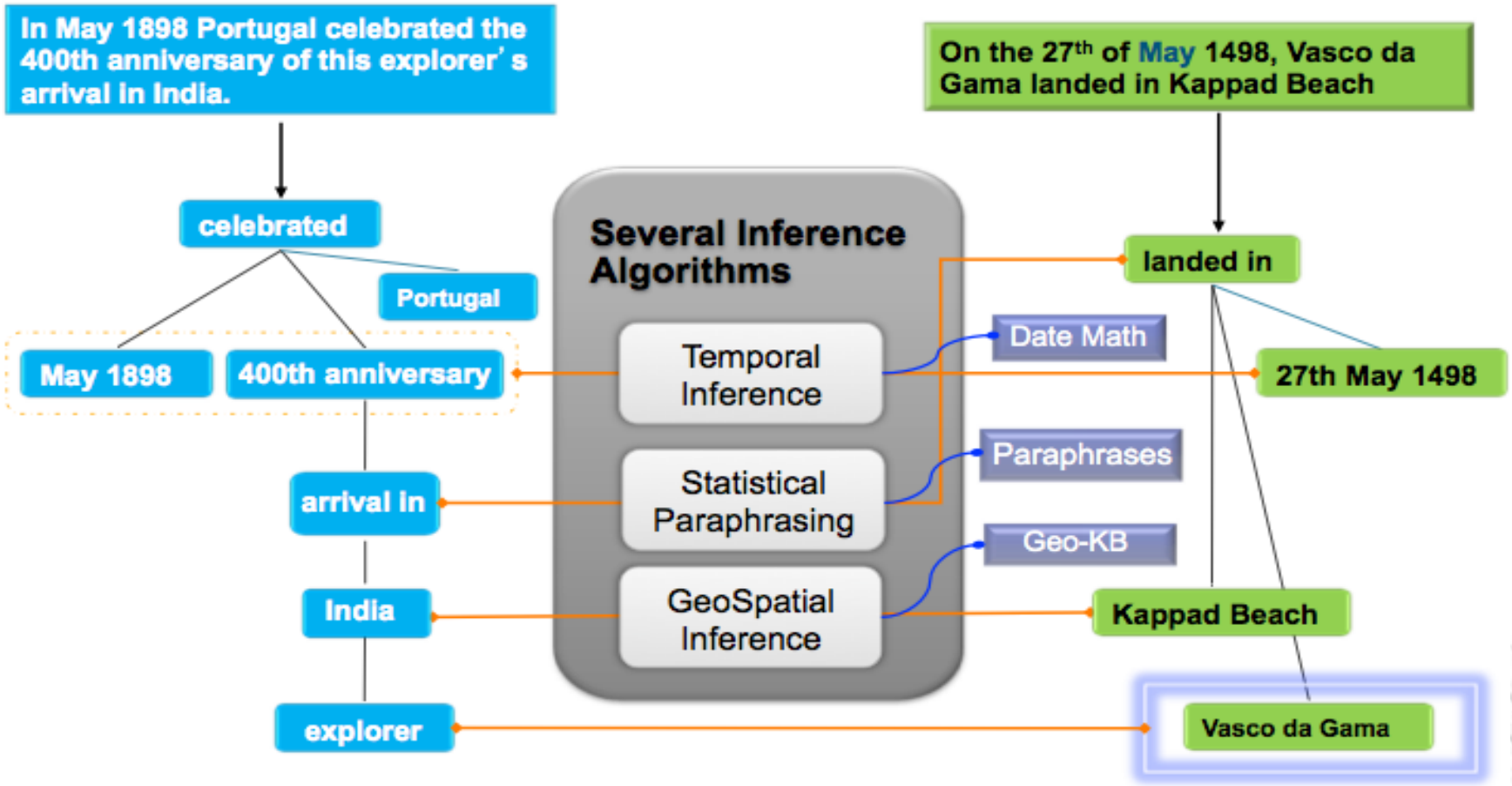
Watson


THINK
DENKE
PENSER
$400
$19,435
$3,400
Ken
WATSON
BRAD
keratin 99%
Porcupine 36%
fur 8%
Inside Edition
NEXT!
21HD

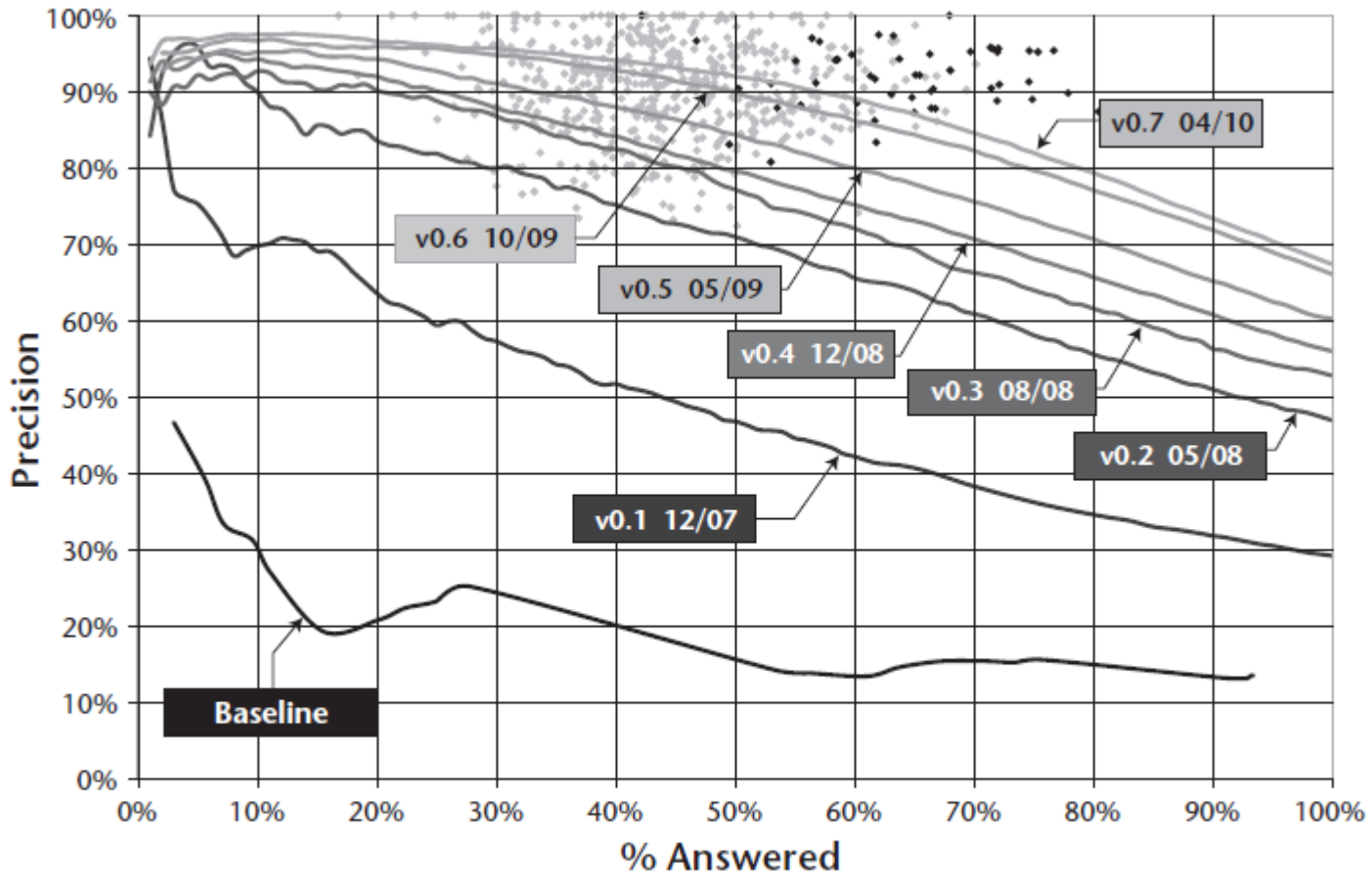
Watson


PIENSE
THINK
ΣΚΕΨΟΥ
DENKE
सोचिए
PENSER
$24,000
Who is Stoker?
$ 1,000
$77,147
Who is Bram Stoker?
$ 17,973
$21,600
WHO IS BRAM STOKER?
$5600

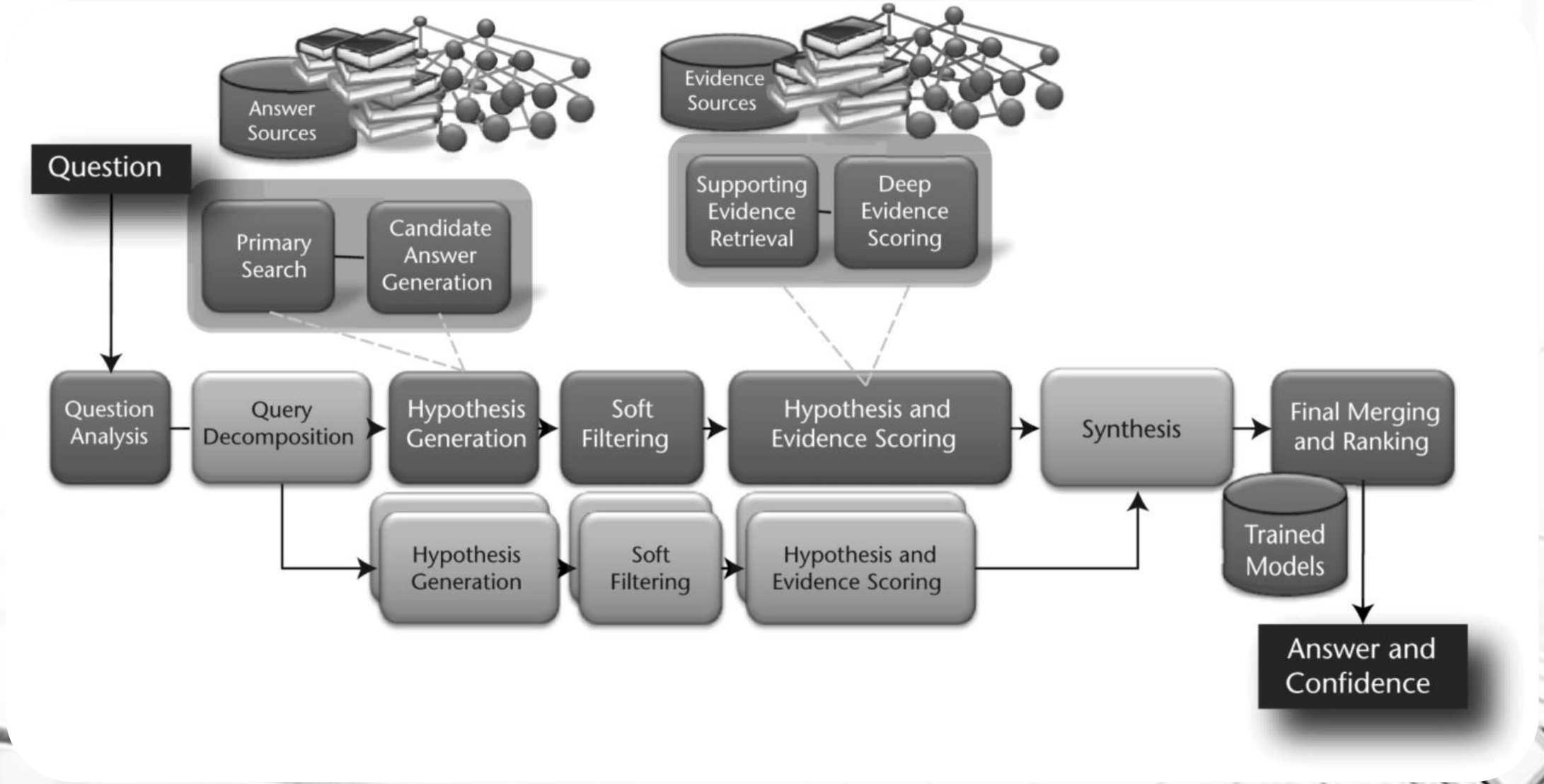
Semantic Inference in Watson QA



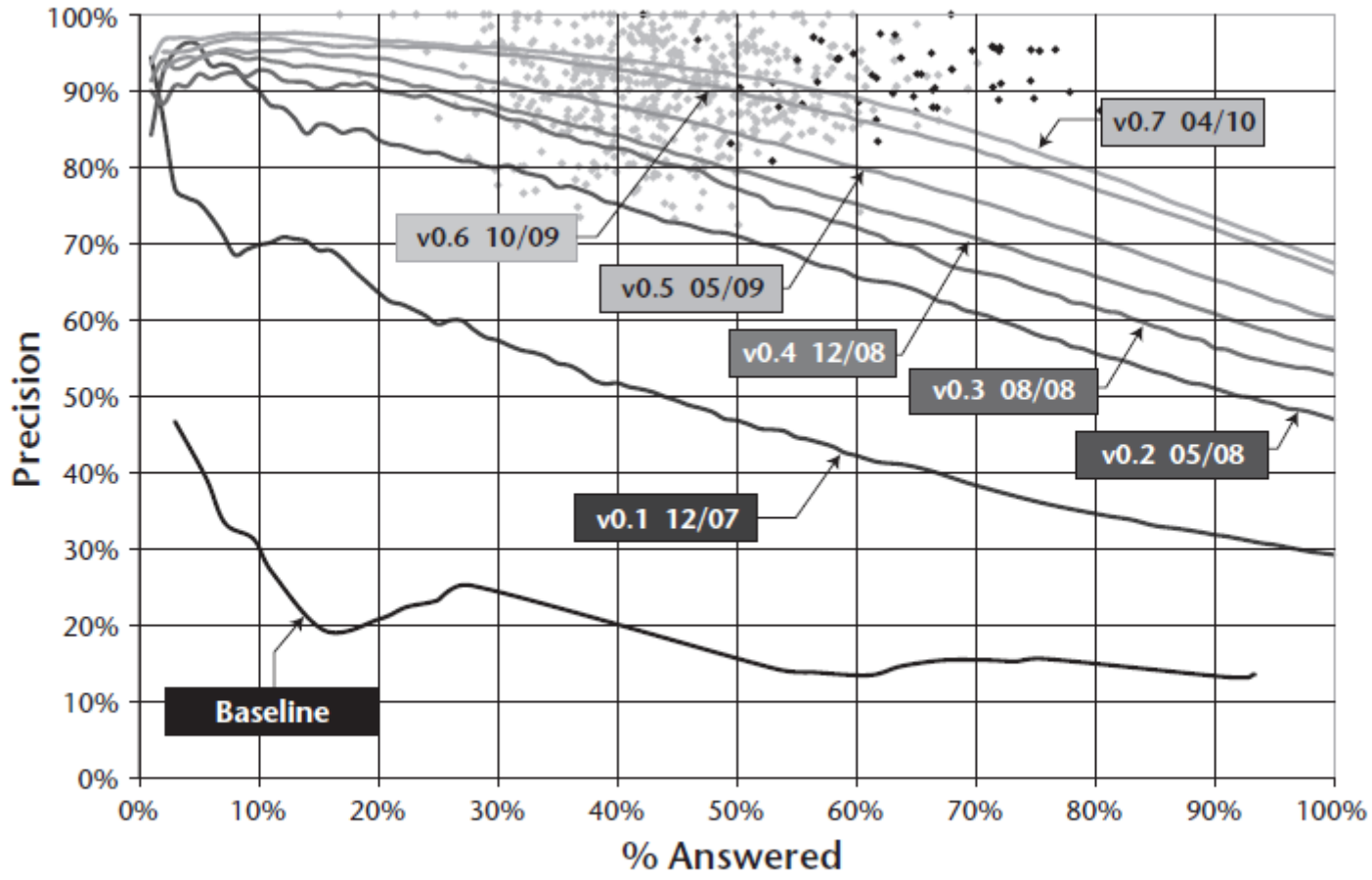
… Intelligence in Watson



Watson: a DeepQA architecture



Ready for Jeopardy!



… about the Watson intelligence

Strongly positive aspects

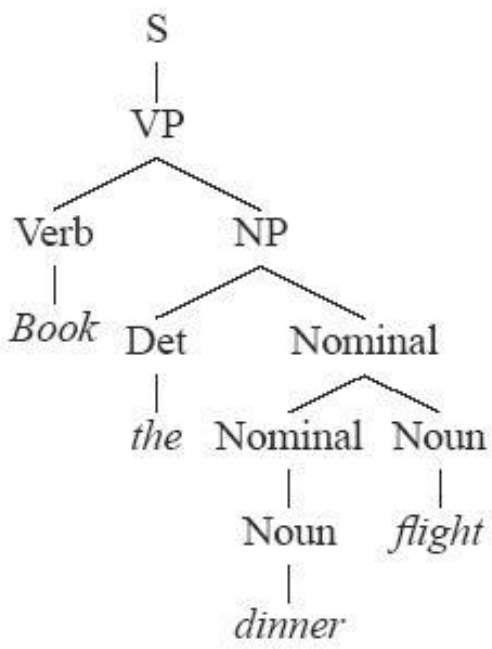
- Adaptivity of the overall workflow
- Significant exploitation of available data
- Huge volumes of knowledge involved

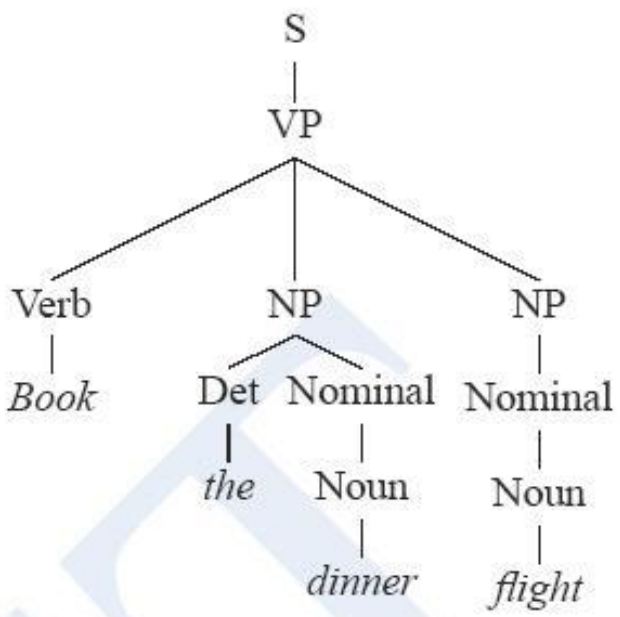
- Criticalities

- The encyclopedic knowledge needed for Jeopardy is quite different in nature from the domain expertise required in many applications
- Wason is based on *Factoid Questions* strongly rooted on objective facts, that are explicit and non subjective
- Formalizing the input knowledge, as it is done a priori for Watson, is very difficult to achive in cost-effective manner: sometimes such knowledge is even absent in an enterprise
- For many natural languages the amount of information and resources is not available, so that a purely data-driven approach is not applicable

Machine Learning: the weapons

- Rule and Patern learning from Data
  - Frequent Pattern Mining (Basket analysis)
- Probabilistic Extensions of Grammars
  - Probabilstic CFGs
  - Stochastic Grammars
- Discriminative learning in neural networks
- SVM: perceptrons
  - Kernel functions in implicit semantic spaces
- Bayesian Models & Graphical Models



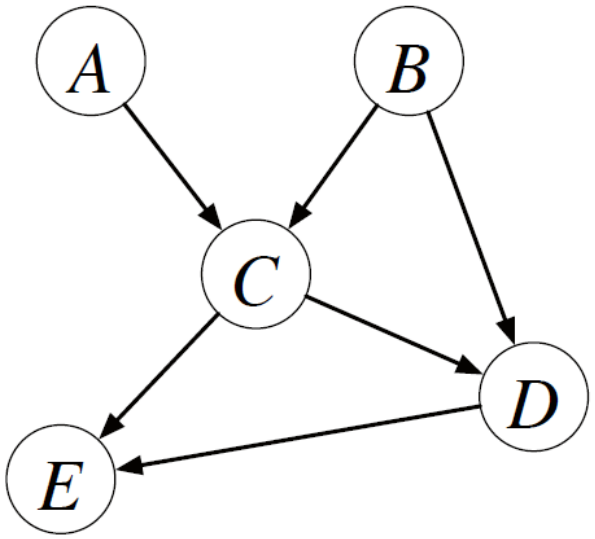


Rules			P
S	→	VP	.05
VP	→	Verb NP	.20
NP	→	Det Nominal	.20
Nominal	→	Nominal Noun	.20
Nominal	→	Noun	.75
Verb	→	book	.30
Det	→	the	.60
Noun	→	dinner	.10
Noun	→	flights	.40

Rules			P
S	→	VP	.05
VP	→	Verb NP NP	.10
NP	→	Det Nominal	.20
NP	→	Nominal	.15
Nominal	→	Noun	.75
Nominal	→	Noun	.75
Verb	→	book	.30
Det	→	the	.60
Noun	→	dinner	.10
Noun	→	flights	.40

Figure 13.2 Two parse trees for an ambiguous sentence, The transitive parse (a) cor-

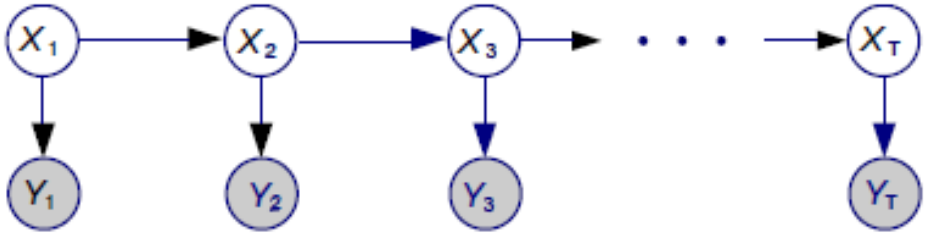
Bayesian & Graphical Models



$$p(A, B, C, D, E) = p(A)p(B)p(C|A, B)p(D|B, C)p(E|C, D)$$

Hidden Markov Models (HMM)

- States = Categories/Classes
- Obesrvations



- Emissions
- Transitions

$$p(X_{1,\ldots,T}, Y_{1,\ldots,T}) = p(X_1)p(Y_1|X_1)\prod_{t=2}^{T}[p(X_t|X_{t-1})p(Y_t|X_t)]$$

- Applications:
  - Speech Recognition
  - Sequence Labeling (e.g. *POS tagging*)

The task of POS tagging

POS tagging

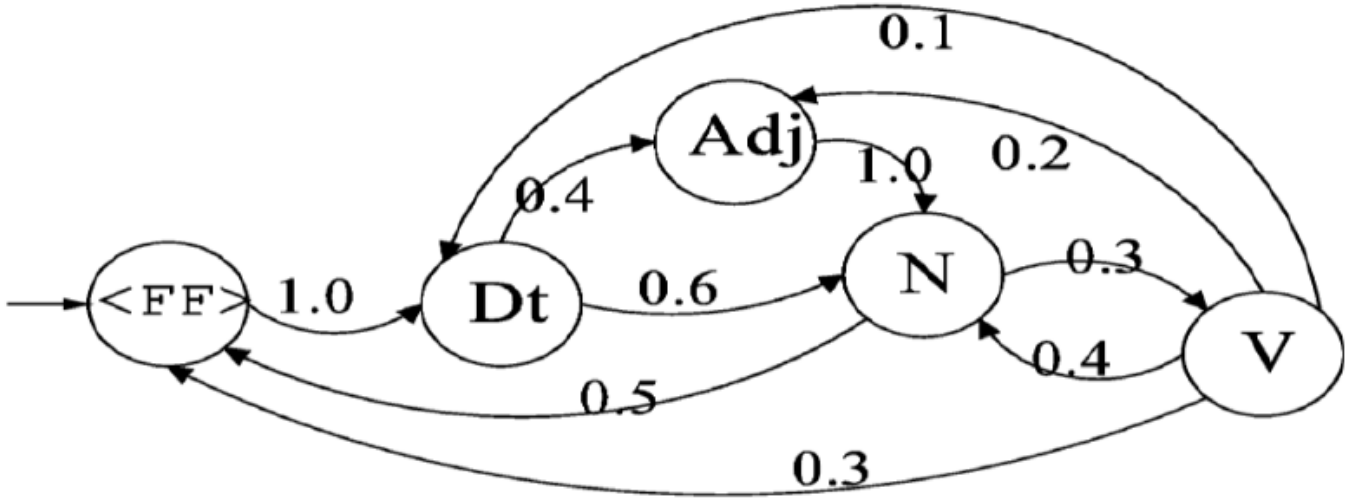
Given a sequence of morphemes $w_1, \ldots, w_n$ with ambiguous syntactic descriptions (i.e.part-of-speech tags) t_j, compute the sequence of n POS tags $t_{j_1}, \ldots, t_{j_n}$ that characterize correspondingly all the words w_i.

Examples:

- *Secretariat is expected to race tomorrow*
- $\Rightarrow$ `NNP VBZ VBN TO VB NR`
- $\Rightarrow$ `NNP VBZ VBN TO NN NR`

The task of POS tagging

An example



Emission probabilities

	.	the	this	cat	kid	eats	runs	fish	fresh	little	big
<FF>	1.0										
Dt		0.6	0.4								
N				0.6	0.1			0.3			
V						0.7	0.3				
Adj									0.3	0.3	0.4

HMM and POS tagging

Given a sequence of morphemes $w_1, \ldots, w_n$ with ambiguous syntactic descriptions (i.e.part-of-speech tags), derive the sequence of n POS tags $t_1, \ldots, t_n$ that maximizes the following probability:

$$P(w_1, \ldots, w_n, t_1, \ldots, t_n)$$

that is

$$(t_1, \ldots, t_n) = argmax_{pos_1, \ldots, pos_n} P(w_1, \ldots, w_n, pos_1, \ldots, pos_n)$$

Note that this is equivalent to the following:
$(t_1, \ldots, t_n) = argmax_{pos_1, \ldots, pos_n} P(pos_1, \ldots, pos_n | w_1, \ldots, w_n)$
as: $\frac{P(w_1, \ldots, w_n, pos_1, \ldots, pos_n)}{P(w_1, \ldots, w_n)} = P(pos_1, \ldots, pos_n | w_1, \ldots, w_n)$
and $P(w_1, \ldots, w_n)$ is the same for all the sequencies $(pos_1, \ldots, pos_n)$.

HMM and POS tagging

How to map a POS tagging problem into a HMM

The above problem

$$(t_1, \ldots, t_n) = argmax_{pos_1,\ldots,pos_n} P(pos_1, \ldots, pos_n | w_1, \ldots, w_n)$$

can be also written (Bayes law) as:

$$(t_1, \ldots, t_n) = argmax_{pos_1,\ldots,pos_n} P(w_1, \ldots, w_n | pos_1, \ldots, pos_n) P(pos_1, \ldots, pos_n)$$

HMM and POS tagging

The HMM Model of POS tagging:

- **HMM States are mapped into POS tags** (t_i), so that $P(t_1, \ldots, t_n) = P(t_1)P(t_2|t_1)\ldots P(t_n|t_{n-1})$
- **HMM Output symbols are words**, so that $P(w_1, \ldots, w_n|t_1, \ldots, t_n) \quad = \quad \prod_{i=1}^{n} P(w_i|t_i)$
- Transitions represent moves from one word to another

Note that *the Markov assumption is used*

- to model probability of a tag in position i (i.e. t_i) only by means of the preceeding part-of-speech (i.e. t_{i-1})
- to model probabilities of words (i.e. w_i) based only on the tag (t_i) appearing in that position (i).

HMM and POS tagging

The final equation is thus:

$$(t_1, \ldots, t_n) = \quad argmax_{t_1,\ldots,t_n} P(t_1, \ldots, t_n | w_1, \ldots, w_n) =$$
$$argmax_{t_1,\ldots,t_n} \prod_{i=1}^{n} P(w_i|t_i) P(t_i|t_{i-1})$$

Fundamental Questions for HMM in POS tagging

1. Given a model what is the probability of an output sequence, O:
   Computing Likelihood.
2. Given a model and an observable output sequence O (i.e. words), how to determine the sequence of states $(t_1, \ldots, t_n)$ such that it is the best explanation of the observation O:
   Decoding Problem
3. Given a sample of the output sequences and a space of possible models how to find out the best model, that is the model that best explains the data:
   how to estimate parameters?

Forward algorithm

In computing the likelihood $P(O)$ of an observation we need to sum up the probability of all paths in a Markov model. Brute force computation is not applicable in most cases.
The forward algorithm is an application of dynamic programming.

The forward algorithm: *estimation*

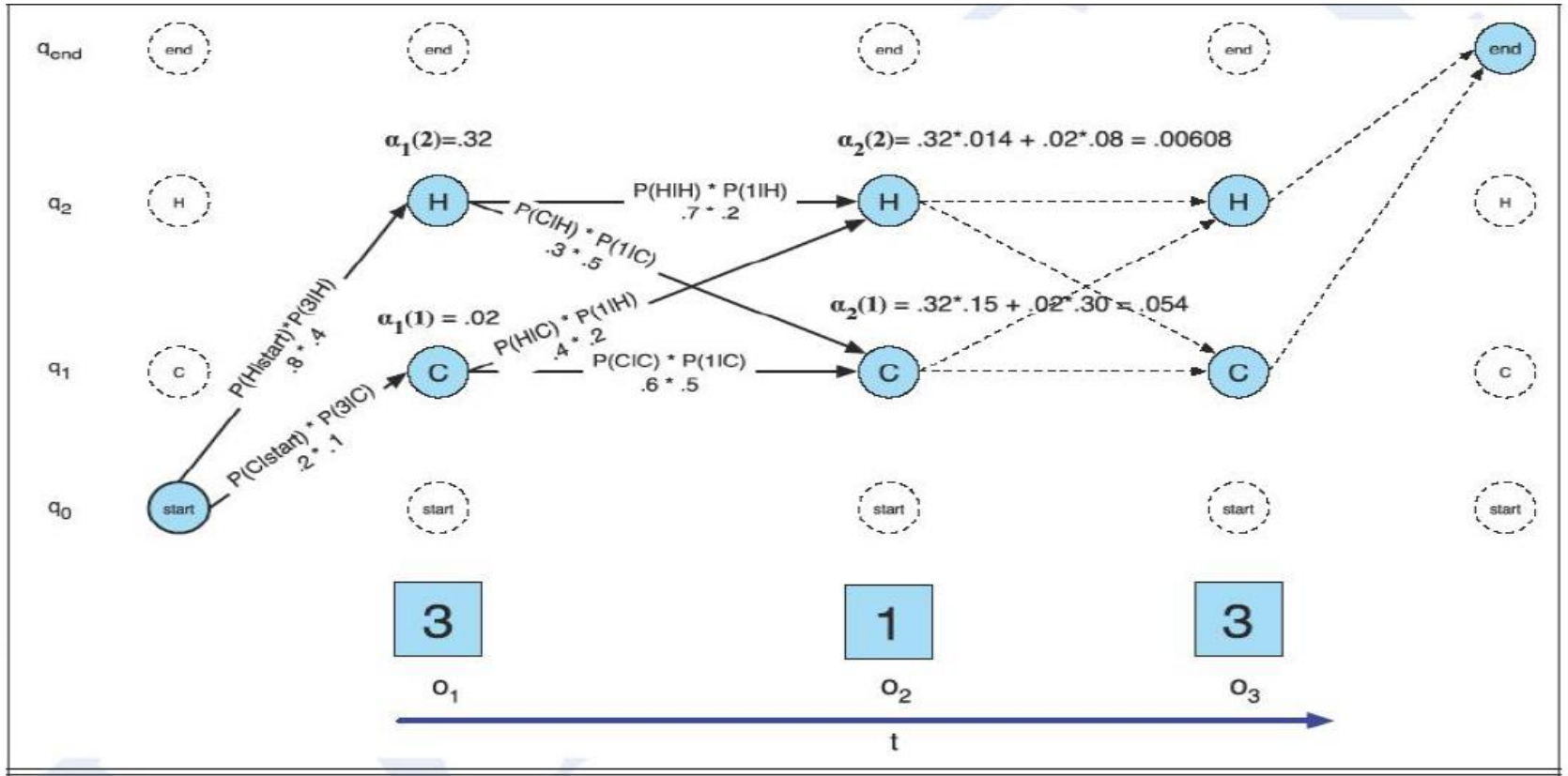
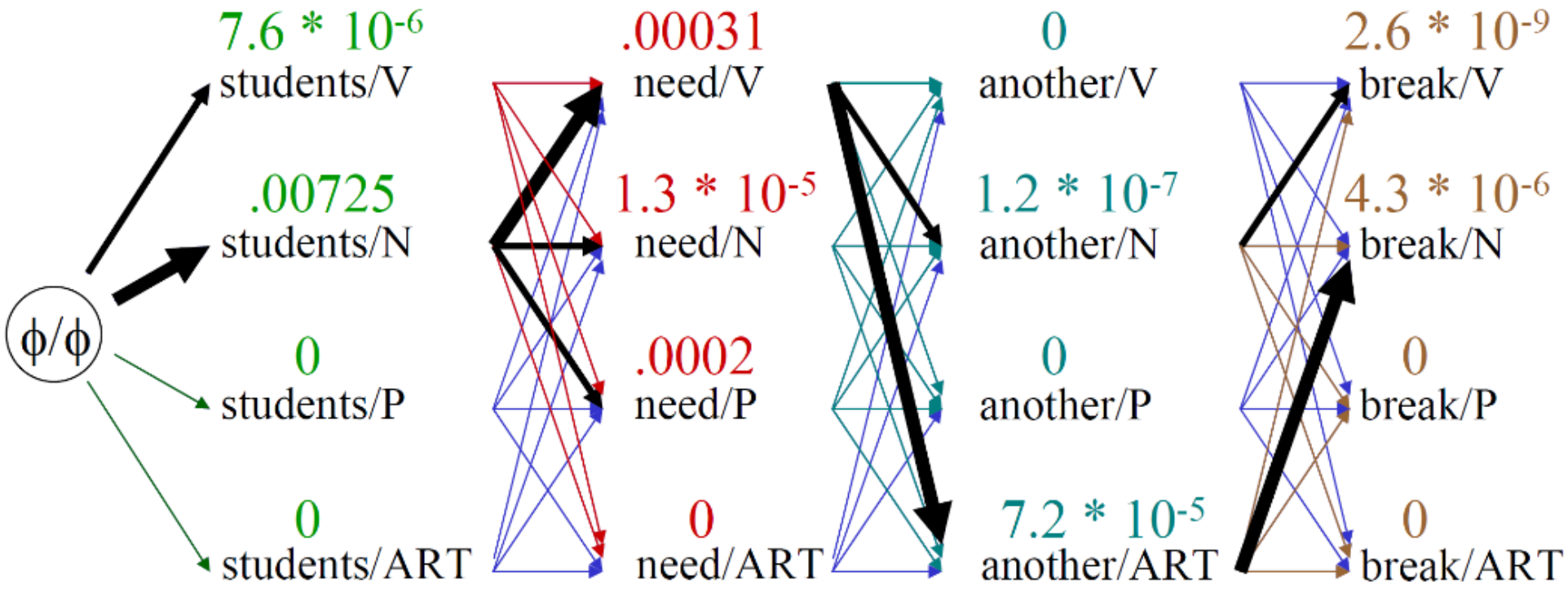


Figure 6.6 The forward trellis for computing the total observation likelihood for the ice-cream events *3 1 3*. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $\alpha_t(j)$ for two states at two time steps. The computation in each cell follows Eq˙ 6.11: $\alpha_t(j) = \sum_{i=1}^{N-1} \alpha_{t-1}(i) a_{ij} b_j(o_t)$. The resulting probability expressed in each cell is Eq˙ 6.10: $\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$.

HMM Decoding: Viterbi Algorithm Intuition:



Viterbi decoding

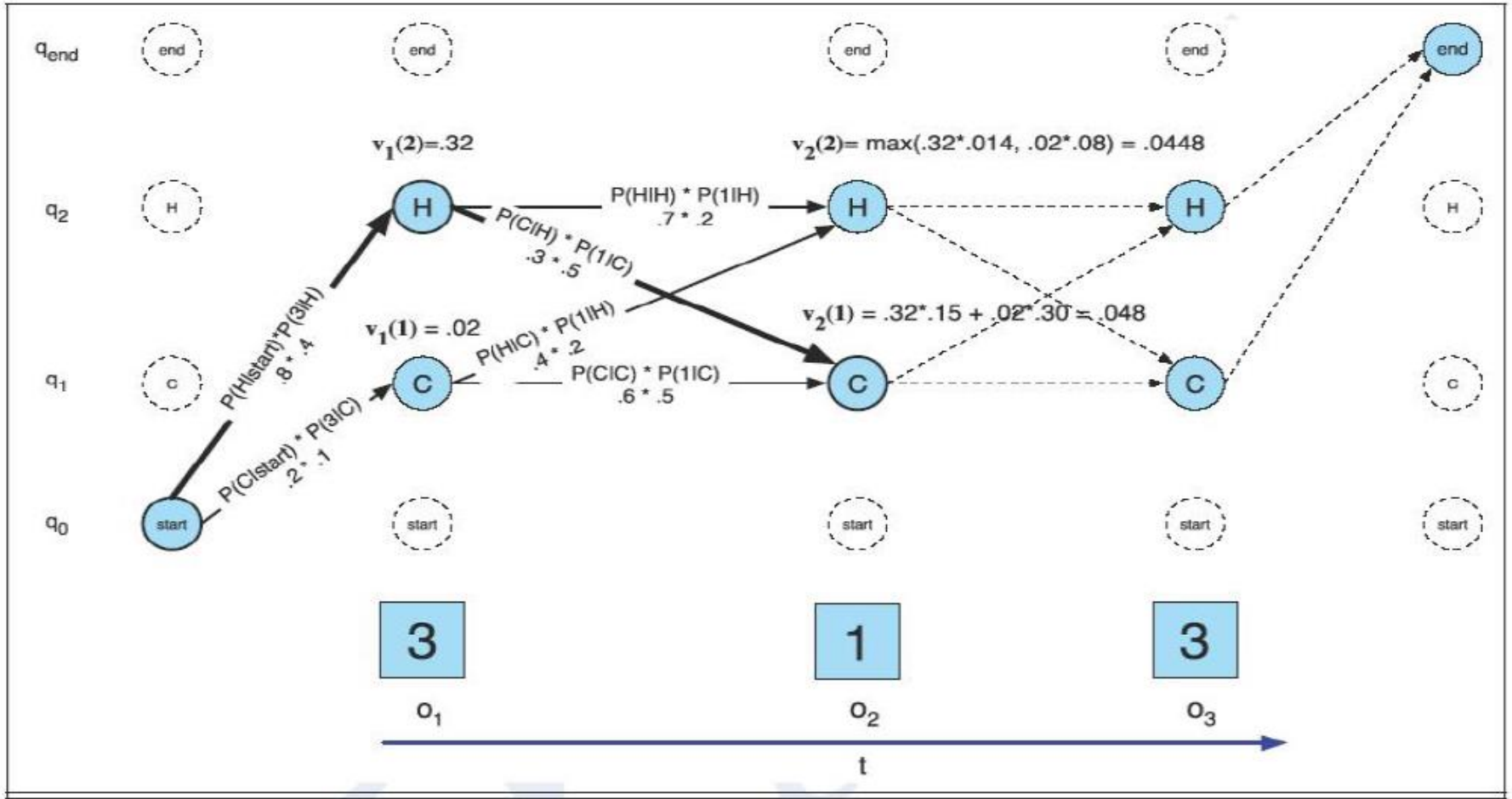


Figure 6.9 The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events *3 1 3*. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $v_t(j)$ for two states at two time steps. The computation in each cell follows Eq˙ 6.10: $v_t(j) = \max_{1 \le i \le N-1} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$ The resulting probability expressed in each cell is Eq˙ 6.16: $v_t(j) = P(q_0, q_1, \ldots, q_{t-1}, o_1, o_2, \ldots, o_t, q_t = j|\lambda)$.

Viterbi decoding

function VITERBI(*observations* of len *T*,*state-graph*) **returns** *best-path*

num-states ← NUM-OF-STATES(*state-graph*)
Create a path probability matrix *viterbi[num-states+2,T+2]*
viterbi[0,0] ← 1.0
for each time step *t* **from** 1 **to** *T* **do**
  for each state *s* **from** 1 **to** *num-states* **do**

$$viterbi[s,t] \leftarrow \max_{1 \le s' \le num\text{-}states} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$$

$$backpointer[s,t] \leftarrow \operatorname*{argmax}_{1 \le s' \le num\text{-}states} viterbi[s',t-1] * a_{s',s}$$

Backtrace from highest probability state in final column of *viterbi[]* and return path

Figure 6.10 Viterbi algorithm for finding optimal sequence of tags. Given an observation sequence and an HMM $\lambda = (A,B)$, the algorithm returns the state-path through the HMM which assigns maximum likelihood to the observation sequence. Note that states 0 and N+1 are non-emitting *start* and *end* states.

NLP and HMM decoding

- The HMM sequence labeling approach can be applied to a variety of linguistic subtasks:
  - Tokenization
  - MWE recognition
  - POS tagging
  - Named Entity Recognition
  - Predicate Argument Structure Recognition
  - SRL: Shallow Semantic Parsing

Multiword Expressions

he was willing to budge a little on

O O O O B b i I

the price which means a lot to me .

O O O B I I I I O

a little; *means a lot to me*; *budge . . . on*

See: "Discriminative lexical semantic segmentation with gaps: running the MWE gamut," Schneider et al. (2014).

Named Entity Recognition

With Commander Chris Ferguson at the helm ,

person

O	B	I	I	O	O	O	O

Atlantis touched down at Kennedy Space Center .

spacecraft location

B	O	O	O	B	I	I	O

Part-of-Speech Tagging

ikr	smh	he	asked	fir	yo	last	name
!	G	O	V	P	D	A	N
interjection	acronym	pronoun	verb	prep.	det.	adj.	noun

so	he	can	add	u	on	fb	lololol
P	O	V	V	O	P	^	!
preposition						proper noun	

Supersense Tagging

ikr	smh	he	asked	fir	yo	last	name
–	–	–	communication	–	–	–	cognition

so	he	can	add	u	on	fb	lololol
–	–	–	stative	–	–	group	–

See: "Coarse lexical semantic annotation with supersenses: an Arabic case study," Schneider et al. (2012).

NLP & Structured Prediction

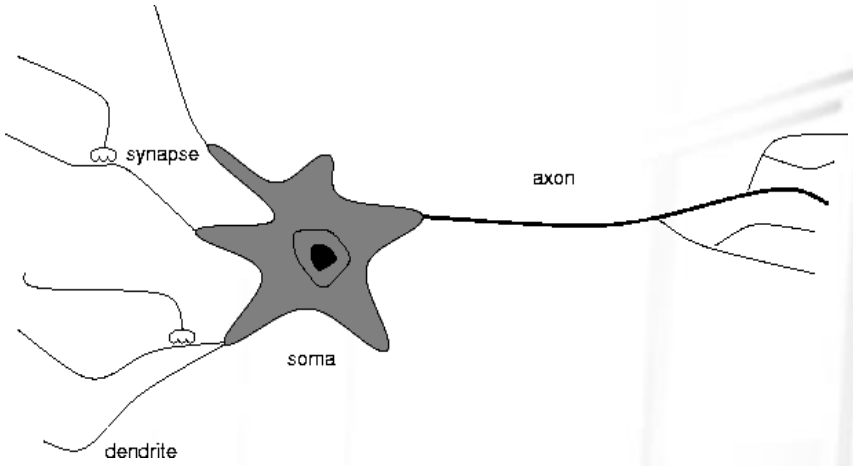
- HMM Decoding is an example of a large class of structured prediction task

- Key elements:
  - Transform a NLP task into into a sequence of classication problem.
  - Transform into a sequence labeling problem and use a variant of the Viterbi algorithm.
  - Design a *representation* (e.g. features and metrics), a *prediction algorithm*, and a *learning algorithm* for your particular problem.

Discriminative Learning

- Characterizes neural networks since the early Cybernetics (Minsky&Papert, 1956)
- Strongly rooted in the notion of
  - Inner product that in turns characterizes the norms thus the distances in the space
  - Use a vector space in R^n as a input representation space
- (not so) Recent Achievments
  - Statistical Learning Theory and Support Vector Machines (Vapnik, 1987)
  - Deep Learning (Bengio et al., 2001)

Linear Classification (1)



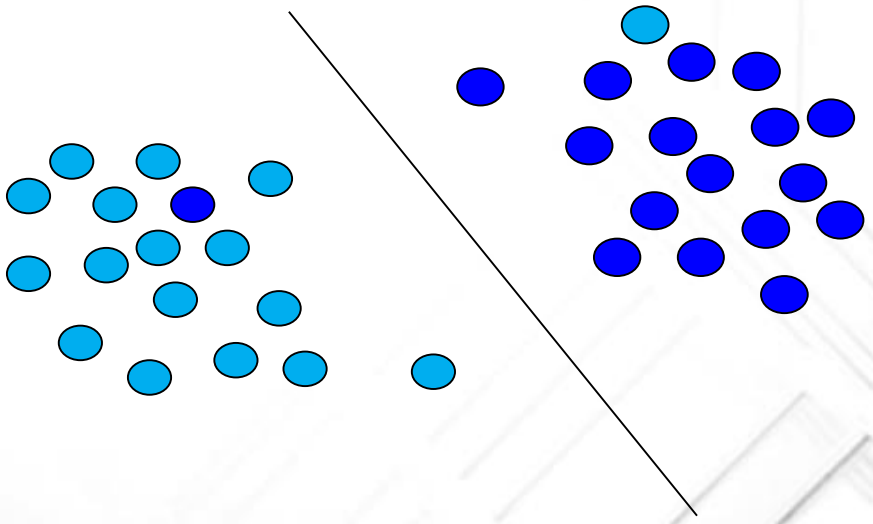
In the hyperplane equation :

$$f(\vec{x}) = \vec{x} \cdot \vec{w} + b, \quad \vec{x}, \vec{w} \in \Re^n, b \in \Re$$

$\vec{x}$ is the vector describing the targeted input example

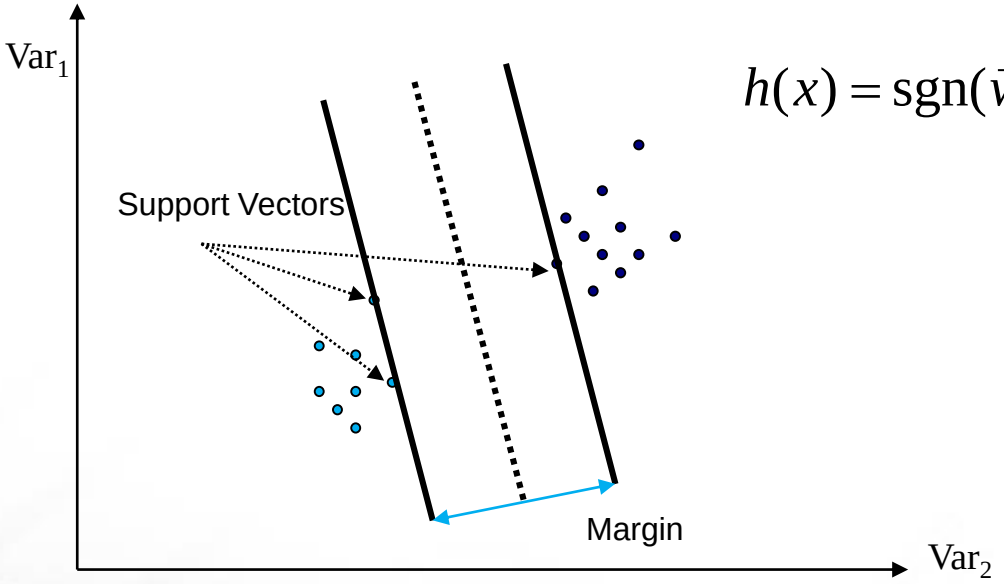
$\vec{w}$ is the gradient of the hyperplane

Classification Inference: $h(x) = \text{sign}(f(x))$



Support Vector Machines

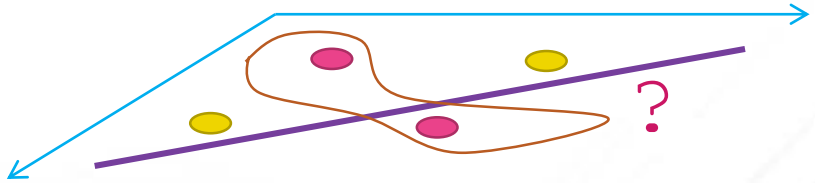
- Support Vector Machines (SVMs) are based on the Statistical Learning Theory [Vapnik, 1995]
  - I t does not require the storage of all data but only a subset of discriminating instances (i.e. the support vectors, SV)
  - The classifier is a linear combination of the SVs (i.e. it depends ONLY on the inner product with their vectors)



$$h(x) = \mathrm{sgn}(\vec{w} \cdot \vec{x} + b) = \mathrm{sgn}(\underbrace{\sum_{j=1..\ell} \alpha_j y_j \vec{x}_j}_{\text{Support Vectors}} \cdot \vec{x} + b)$$

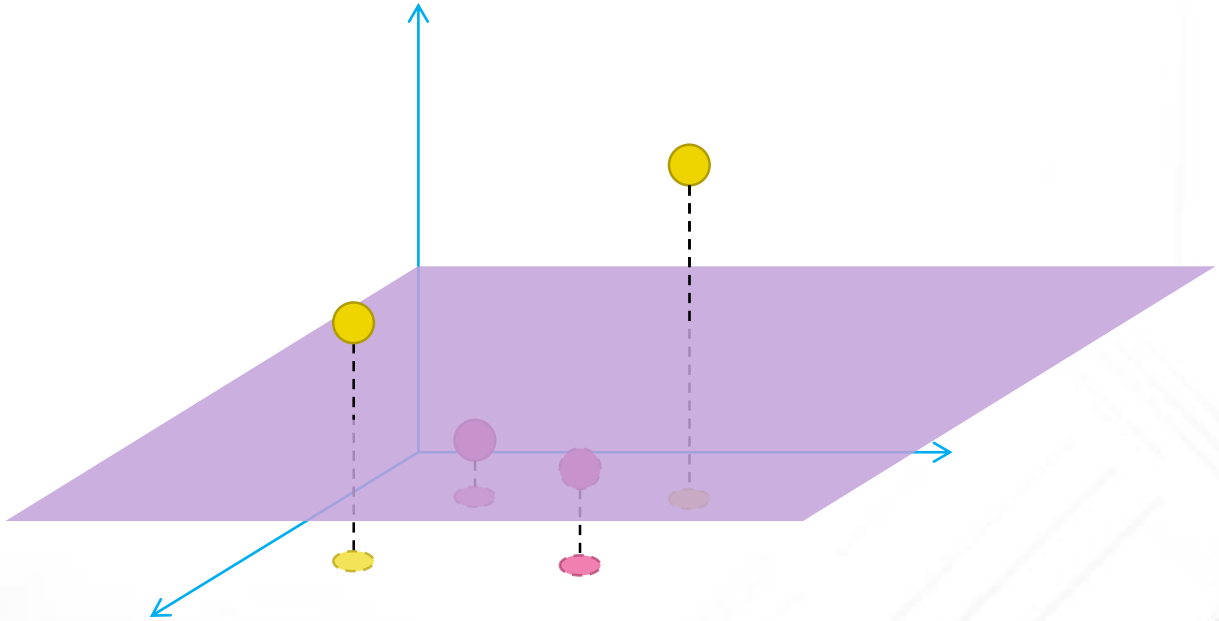
Separability in Higher Dimensional spaces

- In R^2, 3 points can always be separated (or shuttered) by a linear classifier but
  - 4 punti do not (as VC=3) [Vapnik and Chervonenkis(1971)]
- Solution 1 (neural networks): complexify the classification function
  - It needs a more complex architecture usually based on *ensembles* (ad es. multistrato) of neurons
  - Risk of over-fitting on the training data that is dangerous for performance on test ones



Separability and High Dimensional Spaces (2)

- Solution 2: Project instances in an higher dimensional space, i.e. a new feature space by using a projection function ϕ
- Basic idea from SLT: Feature space more complex are preferable to more complex functions as the risk of overfitting is minimized



Representation & Kernels

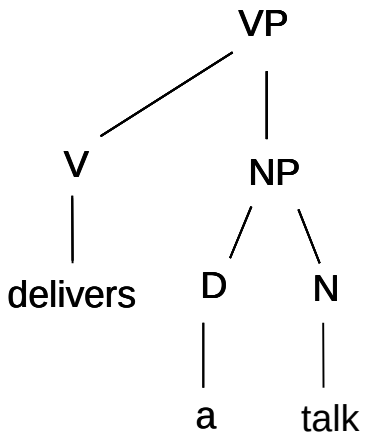
- If a specific function called kernel is available such that $k(xi,xj)=\phi(xi) \cdot \phi(xj)$, there is no need to project the individual examples htrough the projection function ϕ (Cristianini et al., 2002)
- A structured paradigm is applied such that
  - It is trained against more complex structures
  - It moves the machine learning focus onto the representation ($\phi(x_j)$)
- *k(.,.)* expresses a similarity (metrics) that can account for linguistic aspects and depend on the **lexicon**, **sintax** and/or **semantics**

$$h(x) = \mathrm{sgn}(\vec{w} \cdot \varphi(\vec{x}) + b) = \mathrm{sgn}(\overbrace{\sum_{j=1..\ell} \alpha_j y_j \varphi(\vec{x}_j)}^{\text{Support Vectors}} \cdot \varphi(\vec{x}) + b) =$$

$$= \mathrm{sgn}(\sum_{i=1..\ell} \alpha_j y_j k(\vec{x}_j, \vec{x}) + b)$$

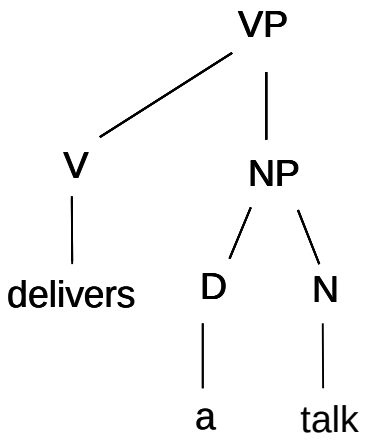
Examples of Kernels sensitive to syntactic structures

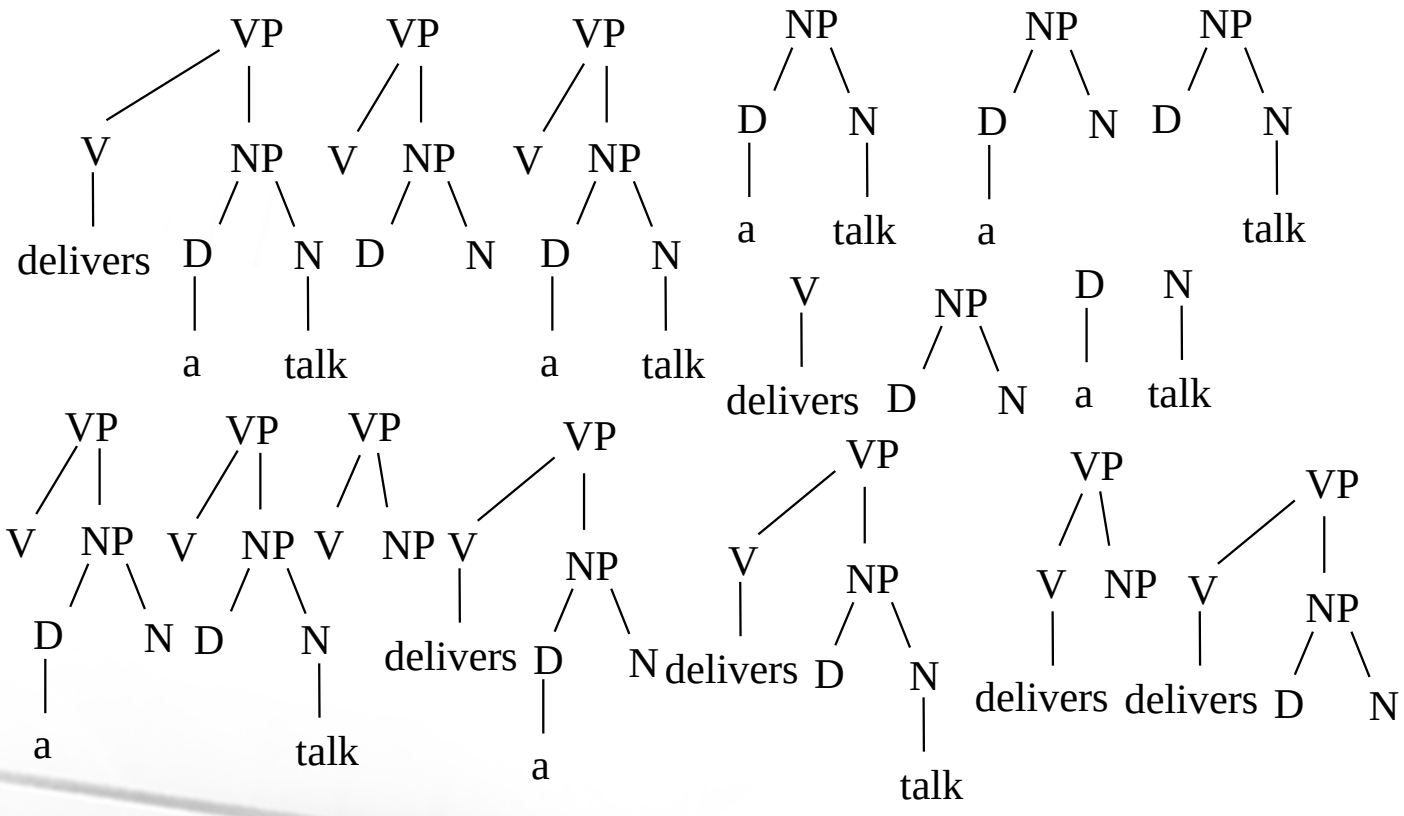
- Given a tree we can see it as the occurrence of a joint event ....



Kernels & Syntactic structures: a collective view of the joint event

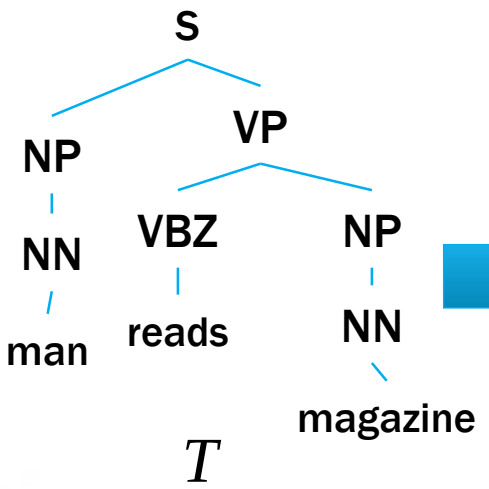
The tree can be see it as the joint occurrence of all the following subtrees:



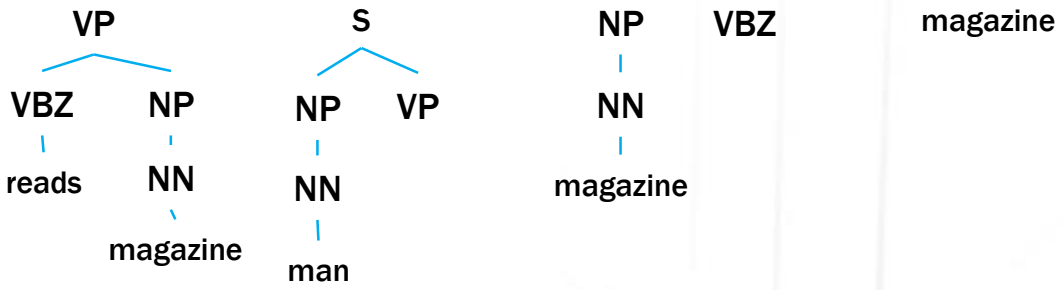


Tree Kernels: the implicit metric space

- The function ϕ in a tree kernel define a vector representing ALL subtrees of the input tree T. It naturally (i.e. without feature engineering) emphasizes:
  - Lexical information (*magazine*)
  - Coarse grain grammatical information (POS tags such as *VBZ*)
  - Syntactic information (frammenti complessi)



$$\varphi(T) = (0,\ldots,1,\ldots,0,\ldots,1,\ldots,0,\ldots,1,\ldots,1,\ldots,0,\ldots,1)$$



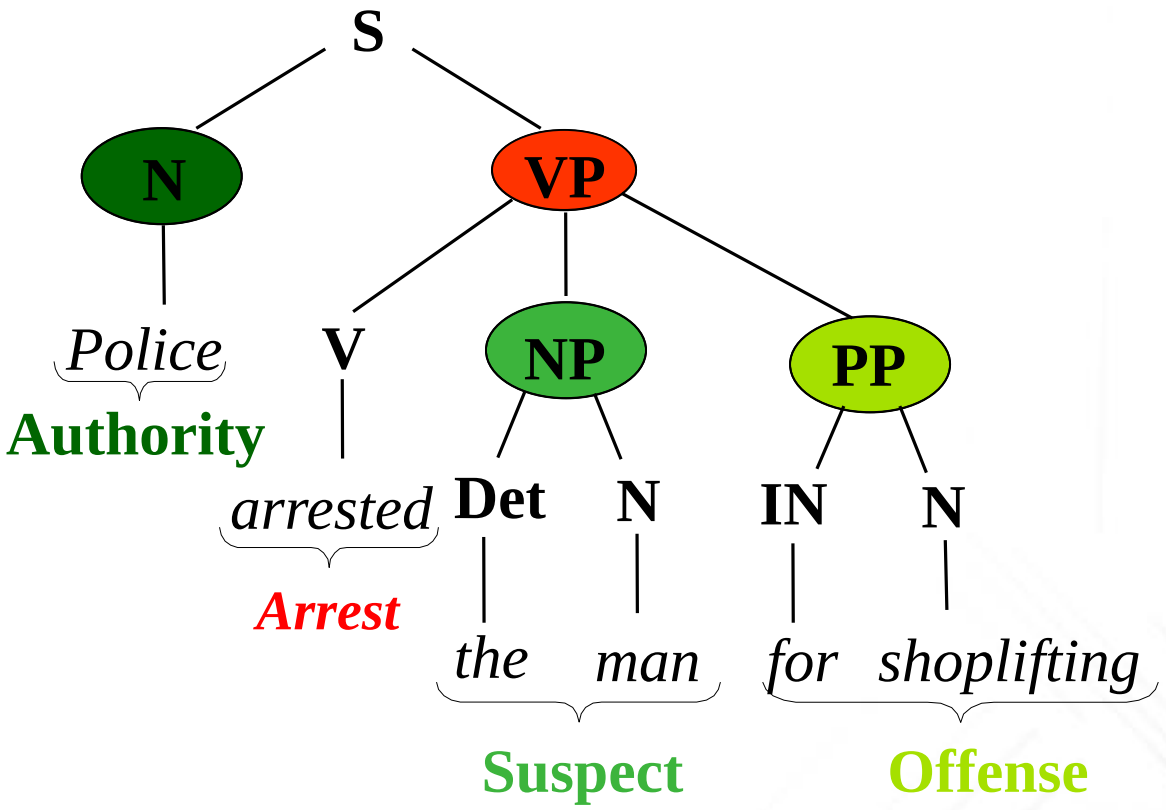
- The inner product in the space of all substrees is proportional to the number of subtrees shared between two sentences
- The learning algorithm (e.g. SVM) will select discriminating examples in (infinite dimensional) space

Application of distributional lexicons for Semantic Role Labeling @ UTV

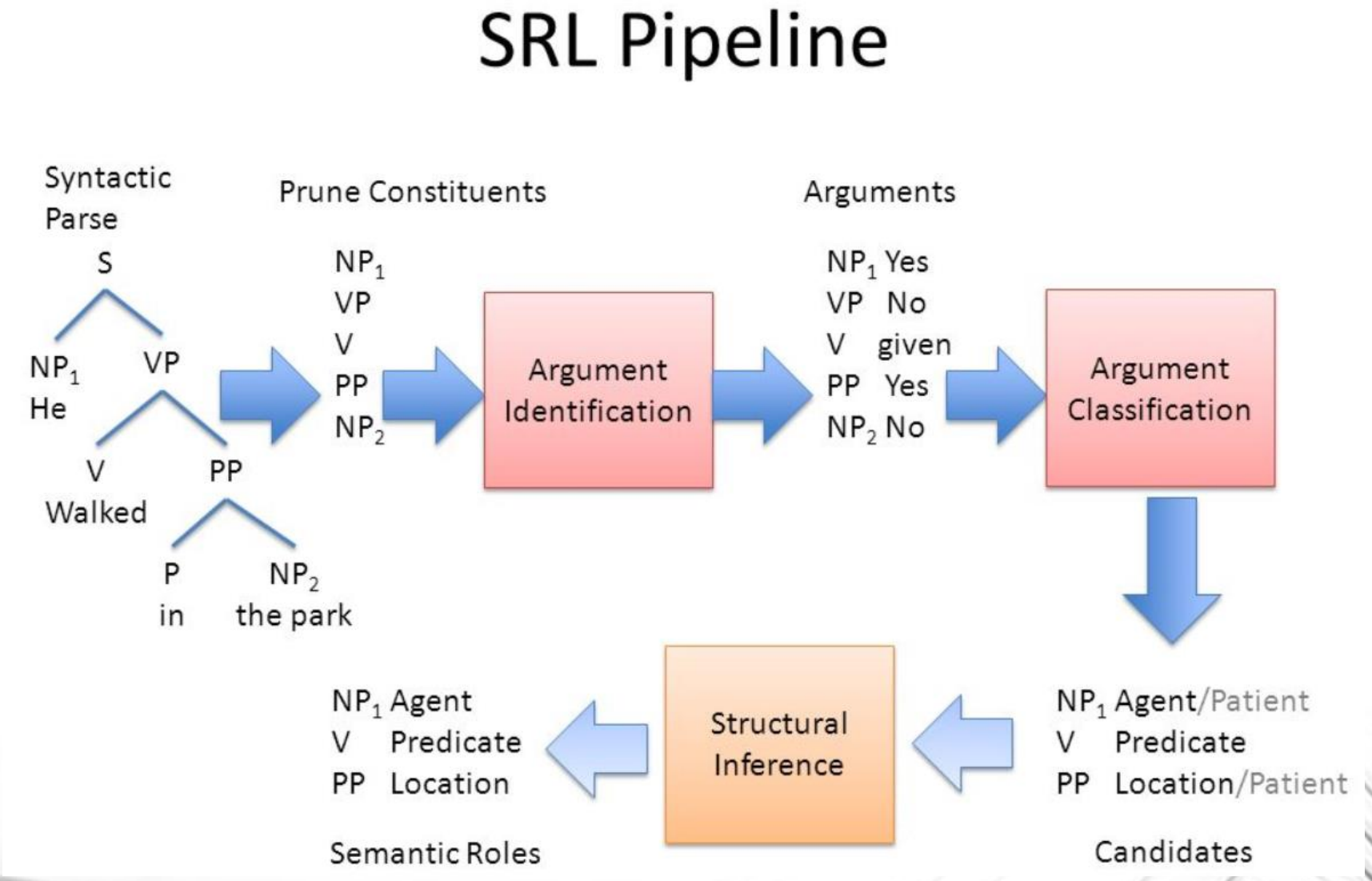
- An important application of tree-kernl based SVMs is Semantic Role labeling wrt Framenet
- In the UTV system, a cascade of classification steps is applied:
  - Predicate detection
  - Boundary recognition
  - Argument categorization (Local models)
  - Reranking (Joint models)
- Input: a sentence and its parse trees
- Adopted kernel: the combination of lexical (e.g. bow) and tree kernel (that is still a kernel)

Linking syntax to semantics

- *Police arrested the man for shoplifting*



Using Framenet/PropBank



Semantic Role Labeling via SVM Learning

- Two steps:
  - Boundary Detection
    - One binary classifier applied to the parse tree nodes
  - Argument Type Classification
    - Multi-classification problem, where n binary classifiers are applied, one for each argument class (i.e. frame element)
    - They are combined in a ONE-vs-ALL scheme, i.e. the argument type that is categorized by an SVM with the maximum score is selected

SRL in Framenet: Results

Eval Setting	Tree Kernels P	R	F_1	Tree Kernels + PK P	R	F_1
				PK alone		
BD	-	-	-	.887	.675	.767
BD Proj.	-	-	-	.850	.647	.735
BD+RC	-	-	-	.654	.498	.565
BD+RC Proj.	-	-	-	.625	.476	.540
		TK			**TK + PK**	
BD	.949	.652	.773	.915	.698	**.792**
BD Proj.	.919	.631	.748	.875	.668	**.758**
BD+RC	.697	.479	.568	.680	.519	**.588**
BD+RC Proj.	.672	.462	.548	.648	.495	**.561**
		TKL			**TKL + PK**	
BD	.938	.659	.774	.908	.701	.791
BD Proj.	.906	.636	.747	.868	.670	.757
BD+RC	.689	.484	.569	.675	.521	.588
BD+RC Proj.	.663	.466	.547	.644	.497	.561

Table 4.1: Results on FrameNet dataset. The table shows Precision, Recall, and F-measure achieved by the Polynomial Kernel (PK) and two different Tree Kernels (TK and TKL). Also, results for their combinations are shown. All experiments exploit 2% training data for Boundary Detection, and 90% for Role Classification.

Framenet SRL: best results

- Best system [Erk&Pado, 2006]
  - **0.855 Precision, 0.669 Recall**
  - **0.751 F1**
- Trento (+RTV) system (Coppola, PhD2009)

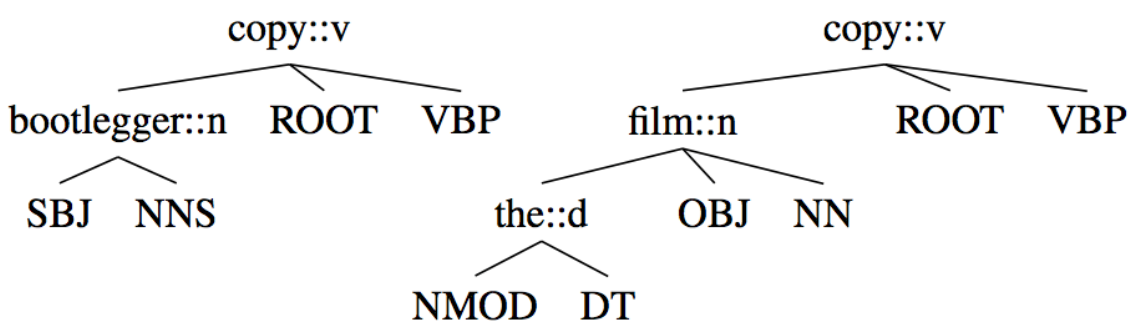
Enhanced PK+TK			
Eval Setting	P	R	F_1
BD (nodes)	1.0	.732	.847
BD (words)	.963	.702	.813
BD+RC (nodes)	.784	.571	.661
BD+RC (words)	.747	.545	.630

Table 4.2: Results on the FrameNet dataset. Best configuration from Table 4.1, raised to 90% of training data for BD and RC.

Argument Classification (Croce et al., 2013)

- UTV experimented with a FrameNet SRL classification (gold standard boundaries)
- We used the FrameNet version 1.3: 648 frames are considered
  - Training set: 271,560 arguments (90%)
  - Test set: 30,173 arguments (10%)

[*Bootleggers*]$_{CREATOR}$, *then* **copy** [*the film*]$_{ORIGINAL}$ [*onto hundreds of VHS tapes*]$_{GOAL}$



Kernel	Accuracy
GRCT	87,60%
$GRCT_{LSA}$	88,61%
LCT	87,61%
LCT_{LSA}	88,74%
GRCT+LCT	87,99%
$GRCT_{LSA}$+LCT_{LSA}	**88,91%**

Semantics, Natural Language & Learning

- From **Learning to Read** to **Knowledge Distillation** as a (integrated pool of) Semantic interpretation Task(s)
  - **Information Extraction**
    - Entity Recognition and Classification
    - Relation Extraction
    - Semantic Role Labeling (Shallow Semantic Parsing)
  - **Estimation of Text Similarity**
    - Structured Text Similarity/Textual Entailment Recognition
    - Sense disambiguation
  - **Semantic Search, Question Classification and Answer Ranking**
  - **Knowledge Acquisition**, e.g. ontology learning
  - **Social Network Analysis, Opinion Mining**

References

- AI & Robotics. «Robot Futures», Ilah Reza Nourbakhsh, MIT Press, 2013
- NLP & ML:
  - «Statistical Methods for Speech Recognition», F. Jelinek, MIT Press, 1998
  - «Speech and Language Processing", D. Jurafsky and J. H .Martin, Prentice-Hall, 2009.
  - "Foundations of Statistical Natural Language Processing, Manning & Schtze, MIT Press 2001.
- URLs:
  - SAG, Univ. Roma Tor Vergata: http://sag.art.uniroma2.it/
  - Reveal s.r.l.: http://www.revealsrl.it/