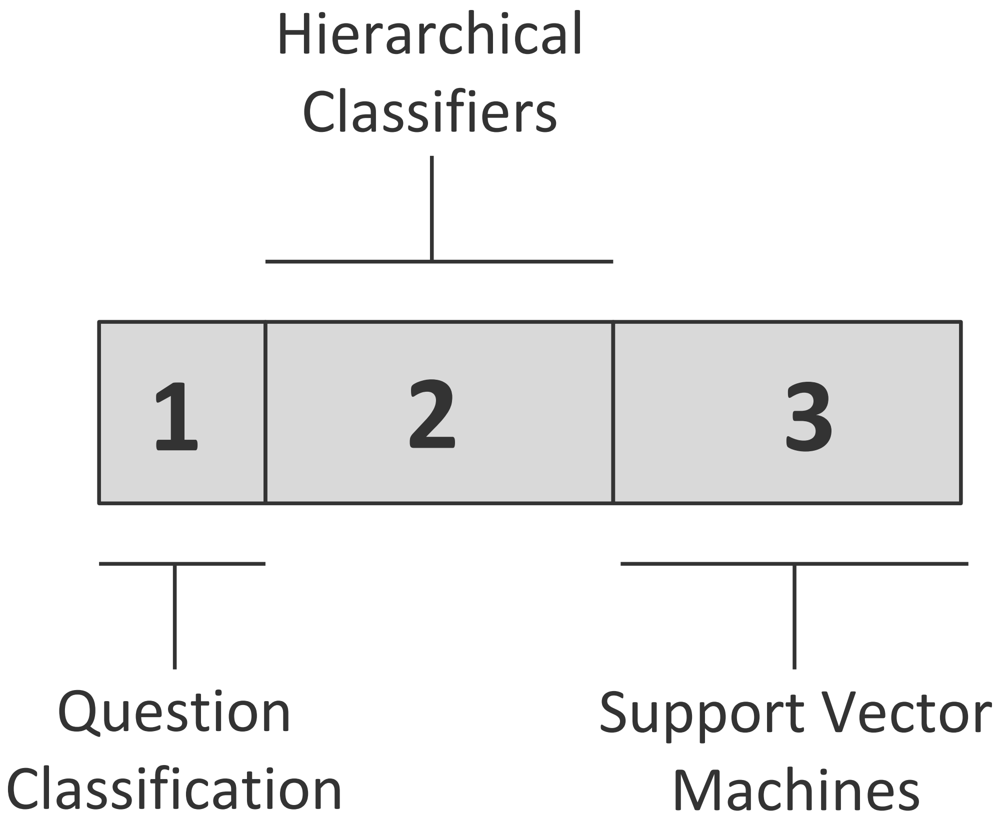
Question Classificaton

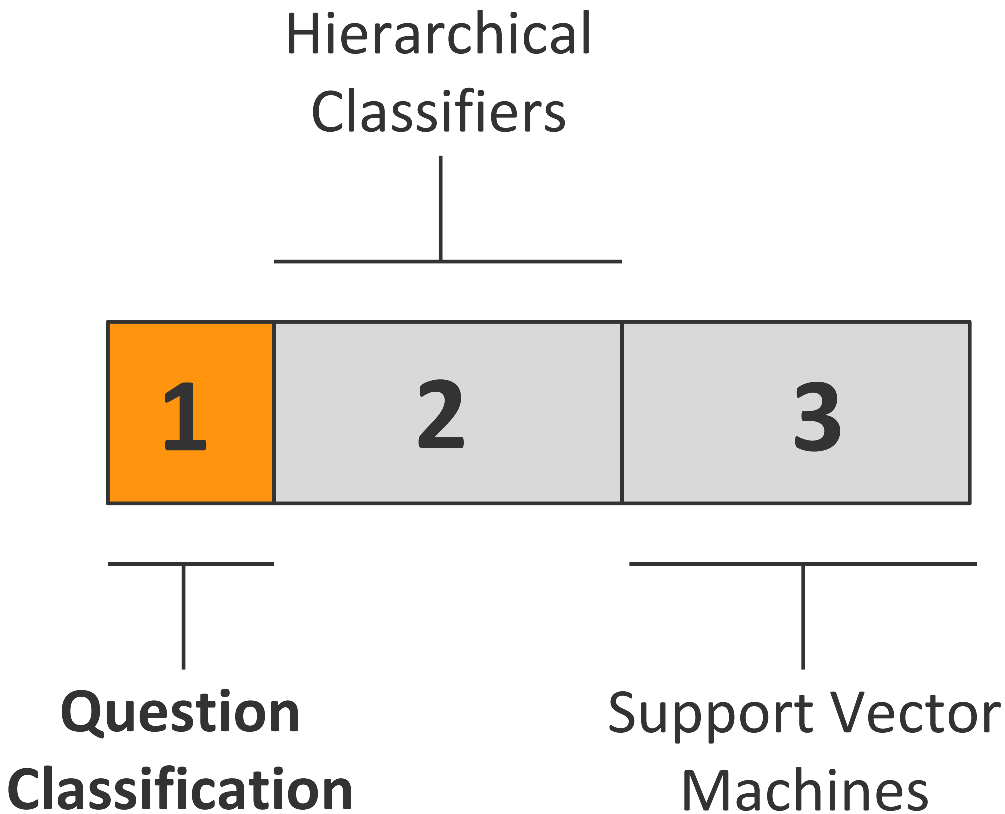
using hierarchical classifiers and support vector machines

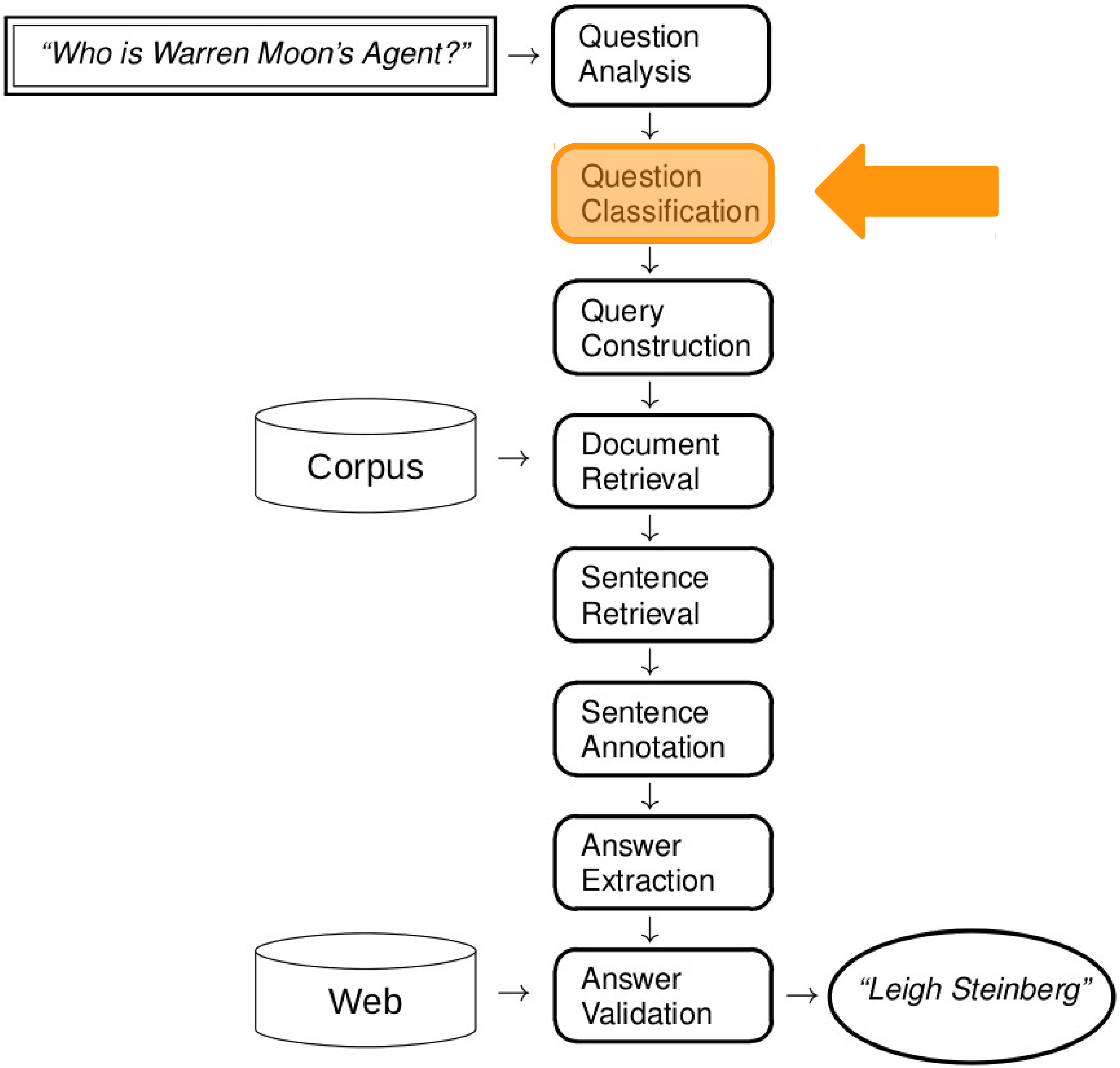
Sebastian Kölle

05. December 2011

Seminar Question Answering


Hierarchical
Classifiers
1
2
3
Question
Classification
Support Vector
Machines


Hierarchical
Classifiers
1
2
3
Question
Classification
Support Vector
Machines



"Who is Warren Moon's Agent?"
Question Analysis
Question Classification
Query Construction
Corpus
Document Retrieval
Sentence Retrieval
Sentence Annotation
Answer Extraction
Web
Answer Validation
"Leigh Steinberg"

Question Classification

„What Canadian city has the largest population?“

→ **LOCATION:city**

„Who was the first man on the moon?“

→ **HUMAN:individual**

„What does 'USA' stand for?“

→ **ABBREV:exp**, **LOCATION:country**

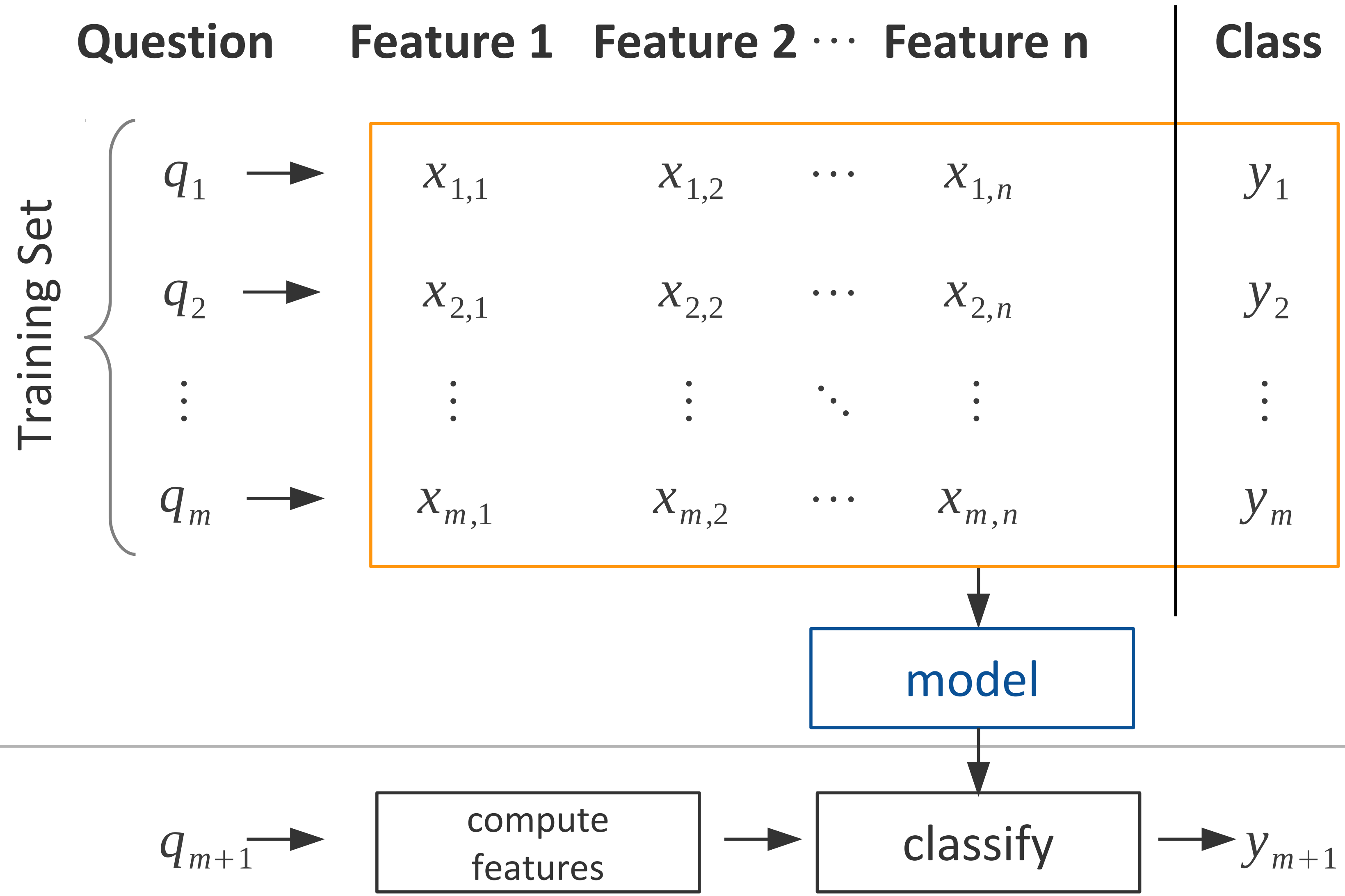
Question Categories

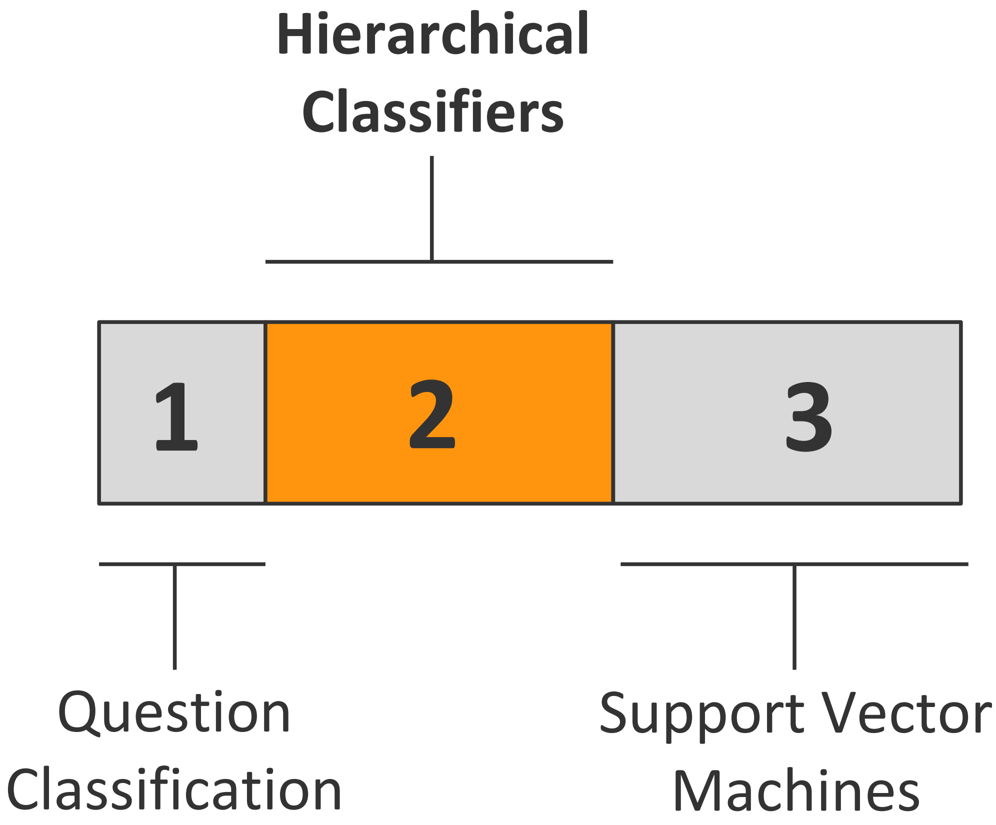
Class	#	Class	#
ABBREV.	9	description	7
abb	1	manner	2
exp	8	reason	6
ENTITY	94	**HUMAN**	65
animal	16	group	6
body	2	individual	55
color	10	title	1
creative	0	description	3
currency	6	**LOCATION**	81
dis.med.	2	city	18
event	2	country	3
food	4	mountain	3
instrument	1	other	50
lang	2	state	7
letter	0	**NUMERIC**	113
other	12	code	0
plant	5	count	9
product	4	date	47
religion	0	distance	16
sport	1	money	3
substance	15	order	0
symbol	0	other	12
technique	1	period	8
term	7	percent	3
vehicle	4	speed	6
word	0	temp	5
DESCRIPTION	138	size	0
defi nition	123	weight	4

Manual classification

Question starts with ...		Class
Who / **Whom**	→	**Person**
Where	→	**Location**
Which / **What**	→	class by **head noun phrase**

Classification and Machine Learning




Hierarchical
Classifiers
1
2
3
Question
Classification
Support Vector
Machines

Approach 1: Classifier

Question features

$\boldsymbol{x_i} = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}]^T$

Input confusion set

e.g. {abbr, entity, desc, human, loc, num}

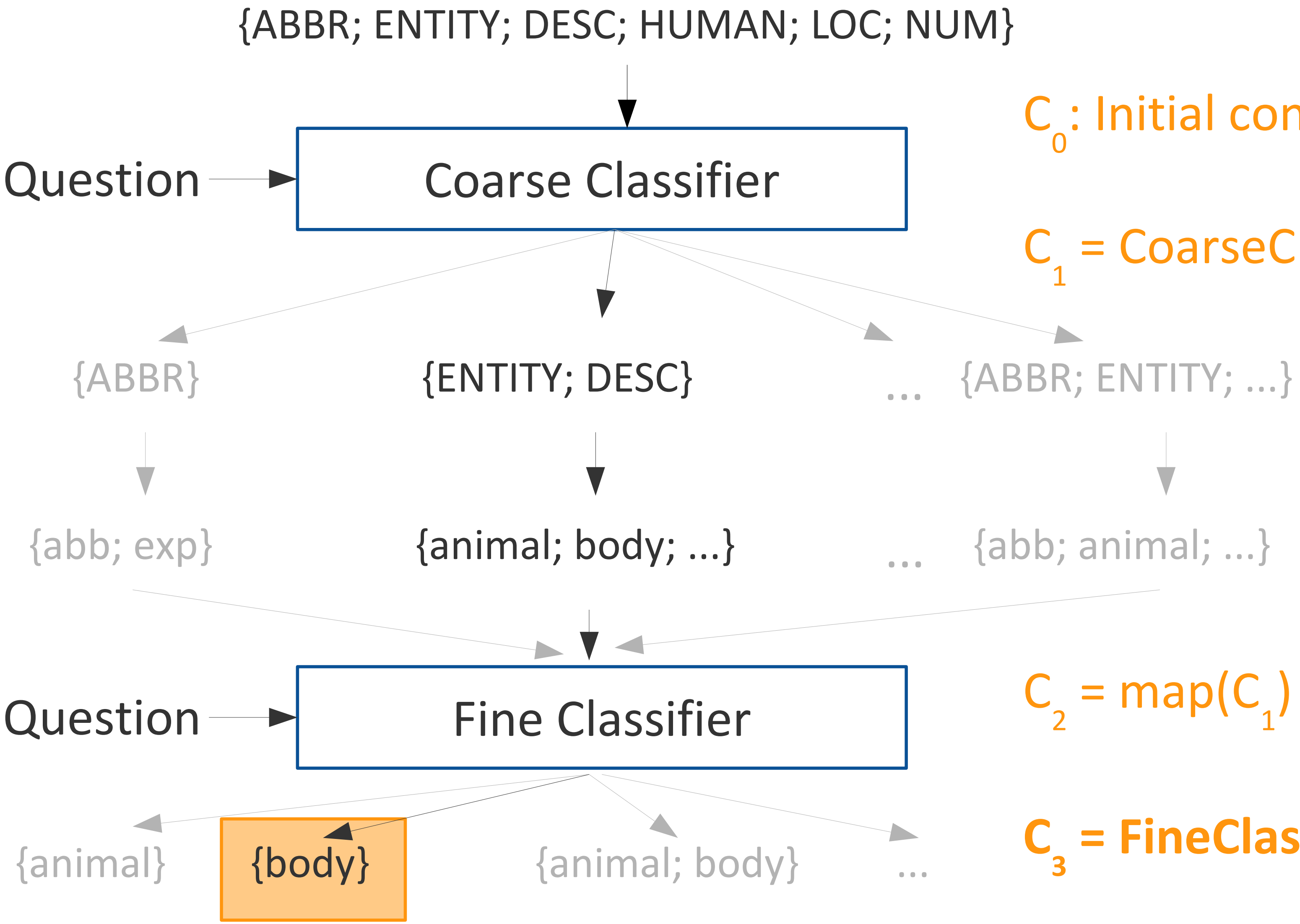
Classifier

1. Compute density for each of the input classes (Winnow algorithm)
2. Sort classes by density
3. Output top k classes (k based on density threshold, max. 5)

Result confusion set

e.g. {entity, desc}

Hierarchical Classifier



C_0: Initial confusion set

C_1 = CoarseClassifier(C_0)

C_2 = map(C_1)

C_3 = FineClassifier(C_2)

Features

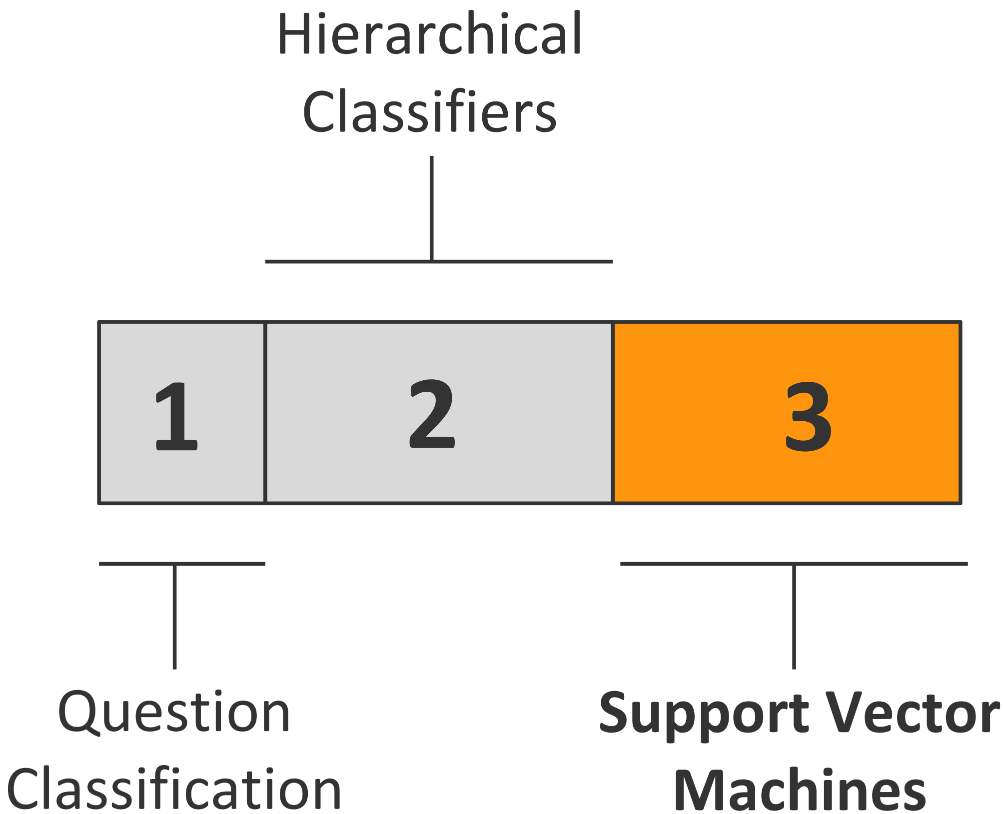
Simple features
(„sensors“)

basic
- words

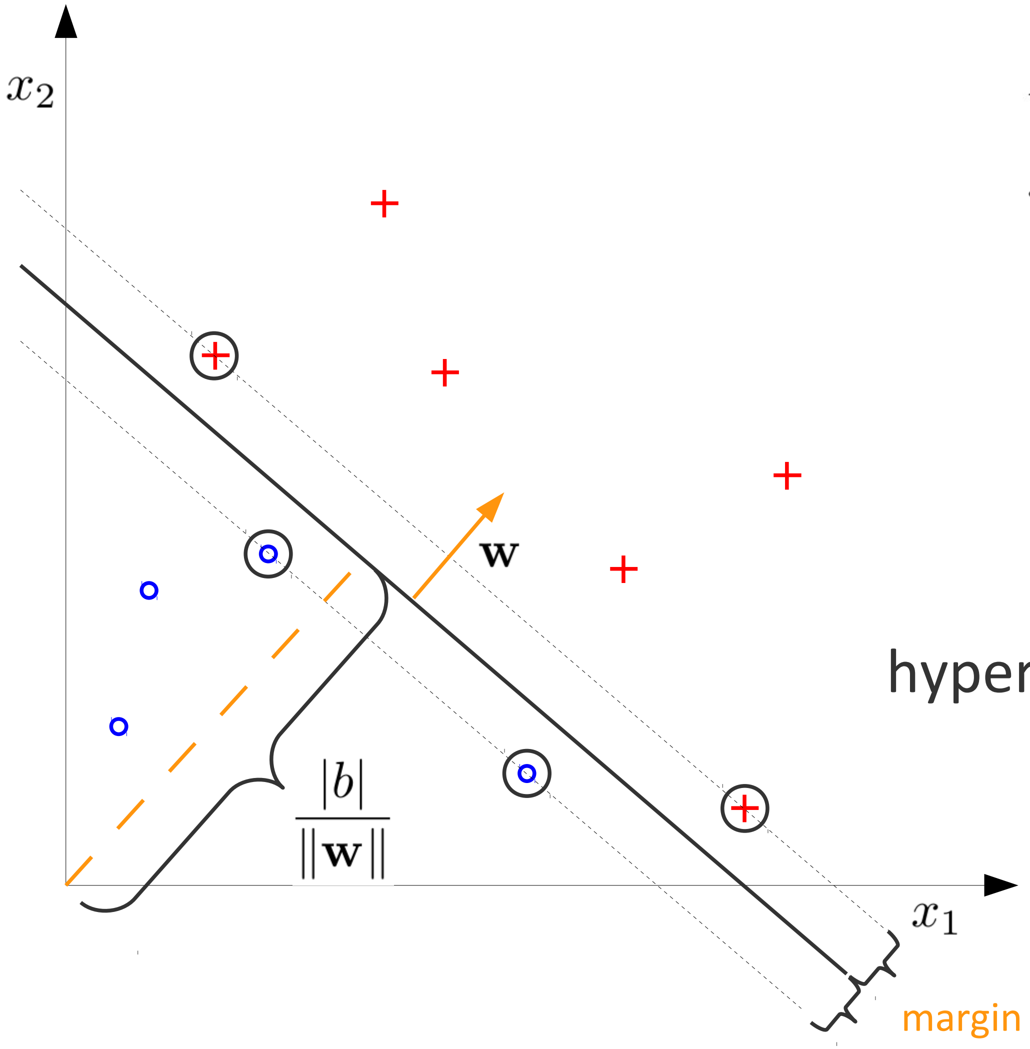
syntactic
- part-of-speech tags
- (head) chunks

semantic
- named entities
- semantically related words

Complex features
- conjunctive (n-grams)
- relational features


Hierarchical Classifiers
1
2
3
Question Classification
Support Vector Machines

Linear Support Vector Machines



$\{(\mathbf{x}_i, y_i); \ldots\}$

$x_i \in \mathbb{R}^2,\ y_i \in \{-1; +1\}$

weights bias

hyperplane $\mathbf{w}^T \cdot \mathbf{x} + b = 0$

$\mathbf{w} \in \mathbb{R}^2,\ b \in \mathbb{R}$

Linear Support Vector Machines

hyperplane $\mathbf{w}^T \cdot \mathbf{x} + b = 0$

1 **training**

find **w**, **b** so the that hyperplane separates the data and the **margin** is maximal

2 **classification**

$$f(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}^T \cdot \mathbf{x} + b \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

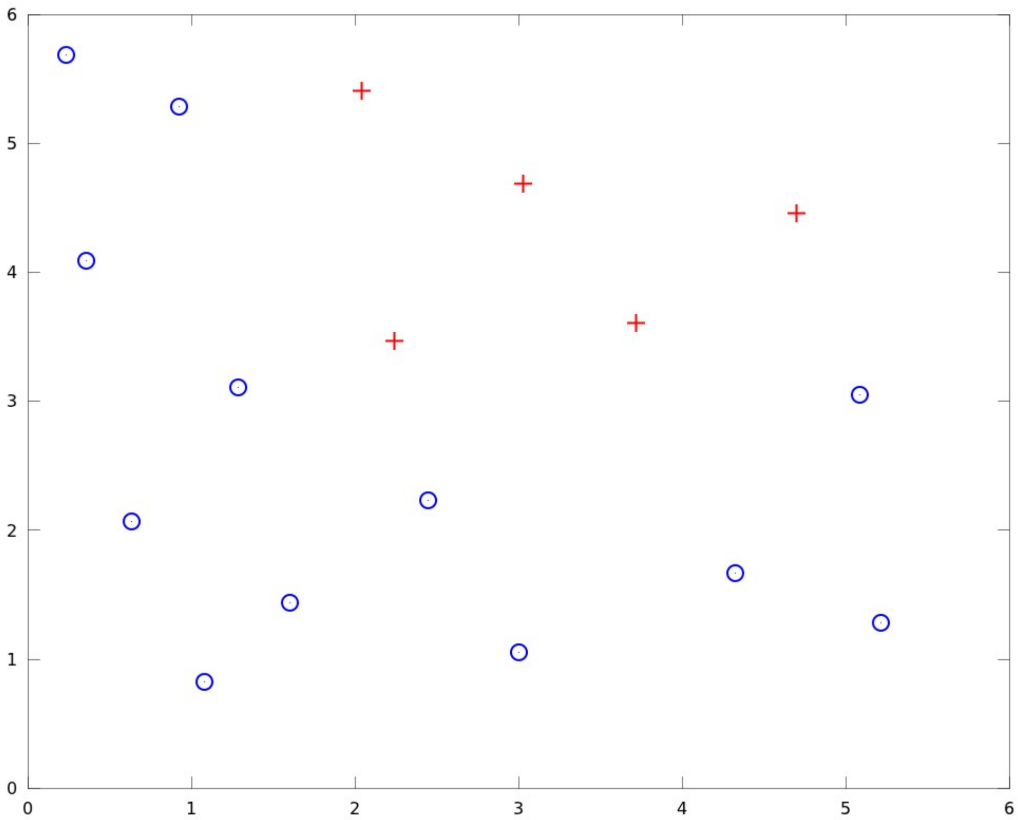
requires **dot product**

(for pairs for feature vectors)

$$\mathbf{x}^T \cdot \mathbf{y} = \sum_{i=0}^{i<d} x_i y_i$$

for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

Nonlinear Support Vector Machines



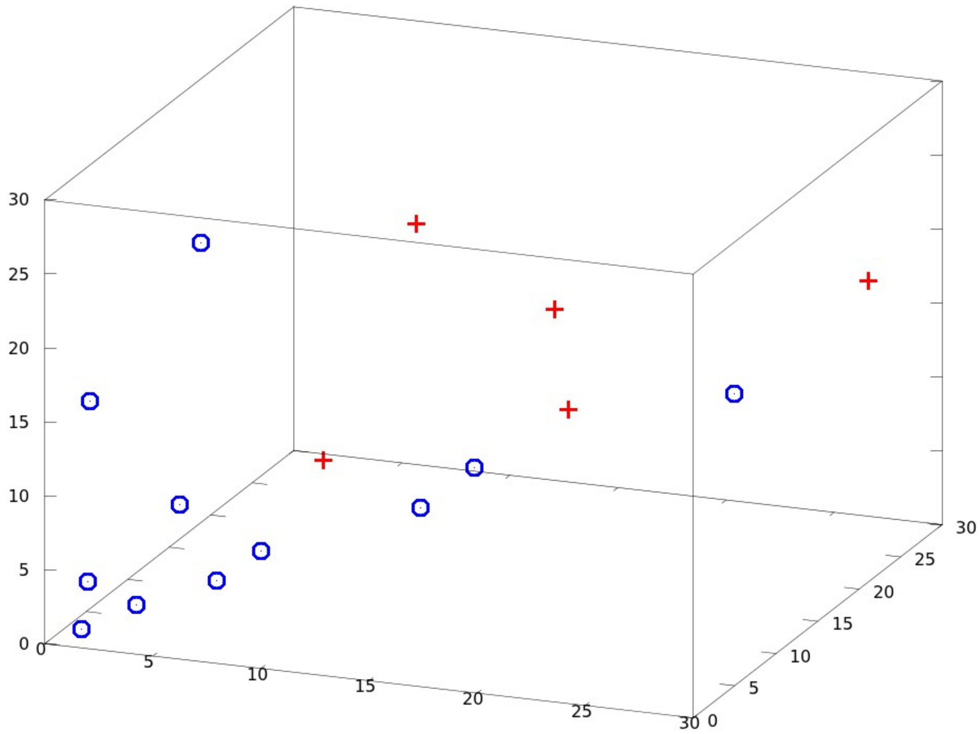
$$\{(\mathbf{x}_i, y_i); \ldots\}$$

$$x_i \in \mathbb{R}^2,\ y_i \in \{\underbrace{-1}_{\circ};\underbrace{+1}_{+}\}$$

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$

$$\Phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

Nonlinear Support Vector Machines



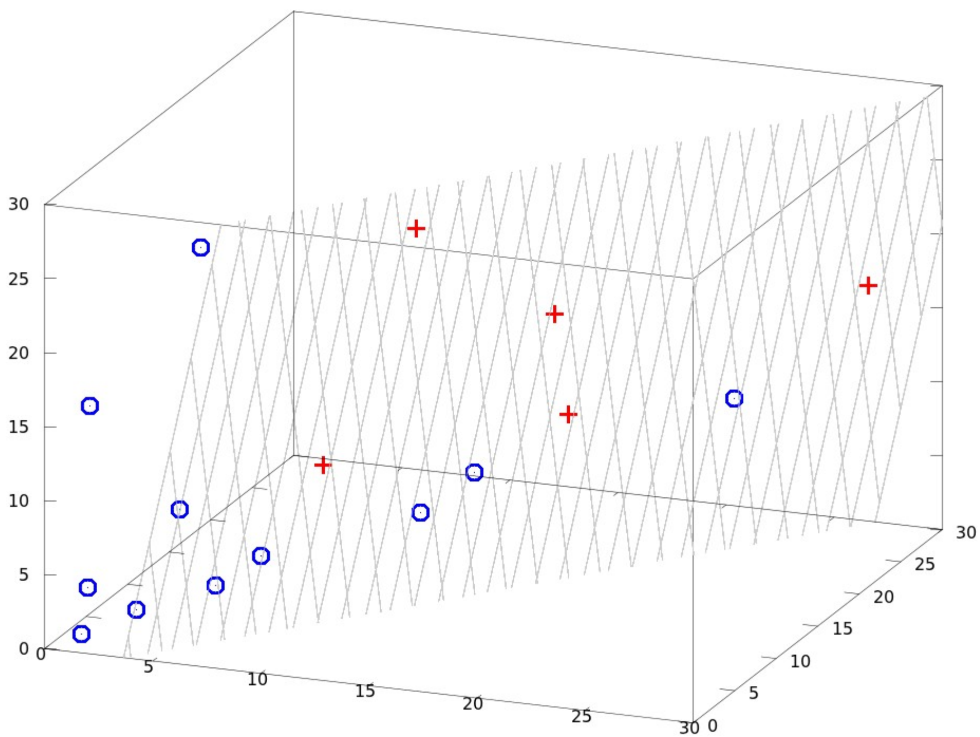
$$\{(\mathbf{x}_i, y_i); \ldots\}$$

$$x_i \in \mathbb{R}^2,\ y_i \in \{\underbrace{-1}_{\circ}; \underbrace{+1}_{+}\}$$

$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$\Phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

Nonlinear Support Vector Machines



$$\{(\mathbf{x}_i, y_i); \ldots\}$$

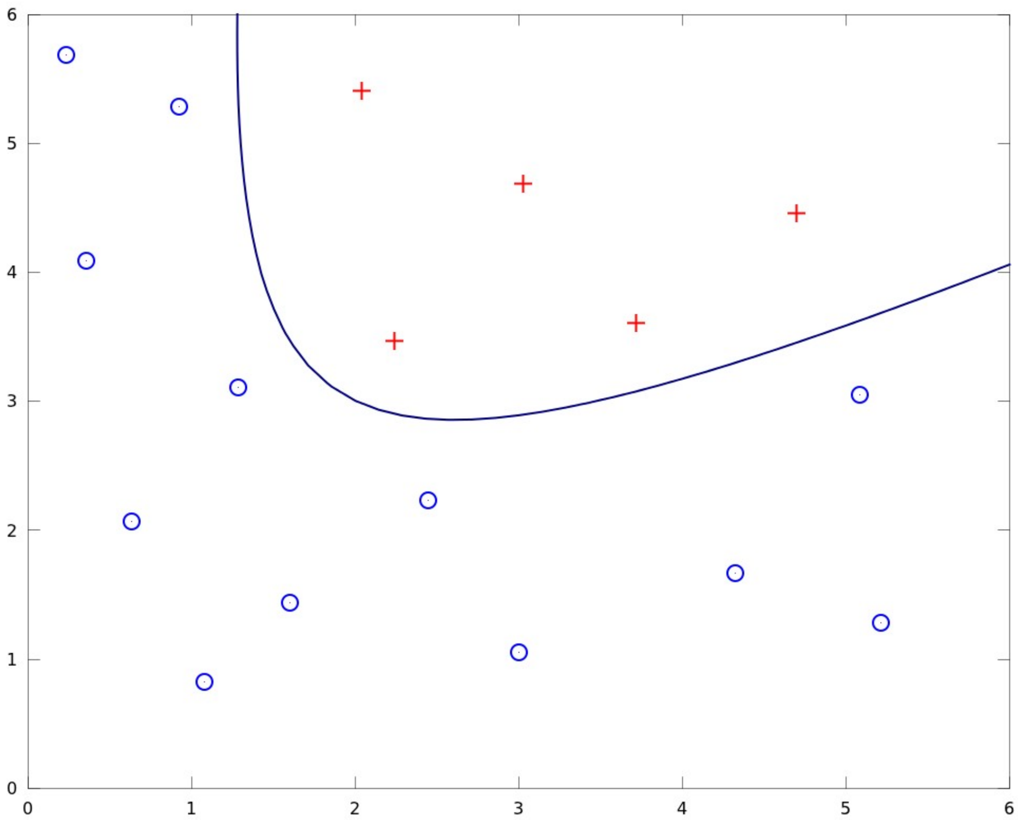
$$x_i \in \mathbb{R}^2,\ y_i \in \{\underbrace{-1}_{\circ}; \underbrace{+1}_{+}\}$$

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$

$$\Phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

$$\mathbf{w}^T \cdot \mathbf{x} + b = 0$$

Nonlinear Support Vector Machines



$$\{(\mathbf{x}_i, y_i); \ldots\}$$

$$x_i \in \mathbb{R}^2,\; y_i \in \{\underbrace{-1}_{\circ};\underbrace{+1}_{+}\}$$

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$

$$\Phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

The Kernel Trick

How to compute this **efficiently**?

- Remember: we need **dot products**

kernel → $K(\mathbf{x},\mathbf{y})$

$\Phi : \mathbb{R}^d \rightarrow \mathcal{H}$ → $\Phi(\mathbf{y})$

$$
\begin{aligned}
K(\mathbf{x},\mathbf{y}) &= \Phi(\mathbf{x})^T \cdot \Phi(\mathbf{y}) \\
&= \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}^T \cdot \begin{bmatrix} y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{bmatrix} \\
&= x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \\
&= (x_1y_1)^2 + 2(x_1y_1)(x_2y_2) + (x_2y_2)^2 \\
&= (x_1y_1 + x_2y_2)^2 \\
&= \boxed{(\mathbf{x}\cdot\mathbf{y})^2}
\end{aligned}
$$

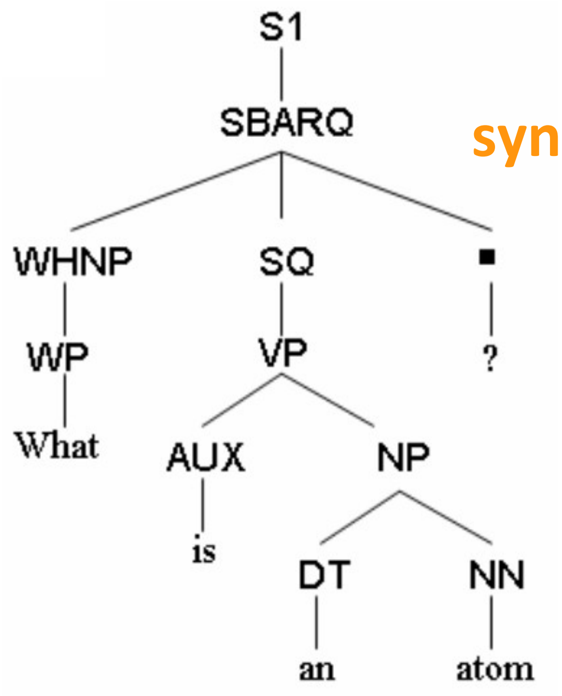
$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$\Phi(\mathbf{x}) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

Features

*„**Which univerity** did **the president graduate** from?“*

*„**Which president** is a **graduate** of **the** Harvard **University**?“*

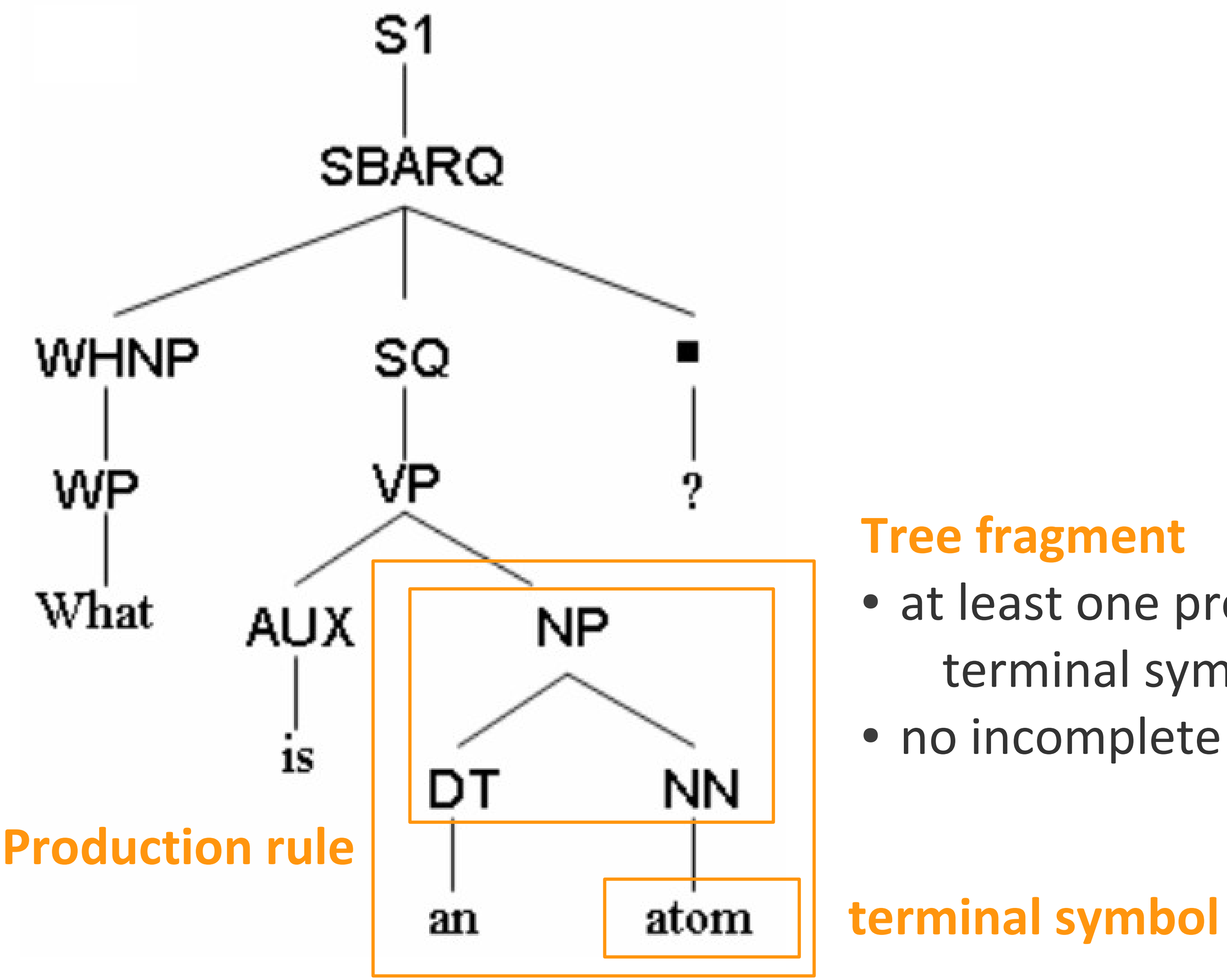


syntax tree

tree kernel

$$K(T_1, T_2) = \boldsymbol{v}(T_1) \cdot \boldsymbol{v}(T_2)$$

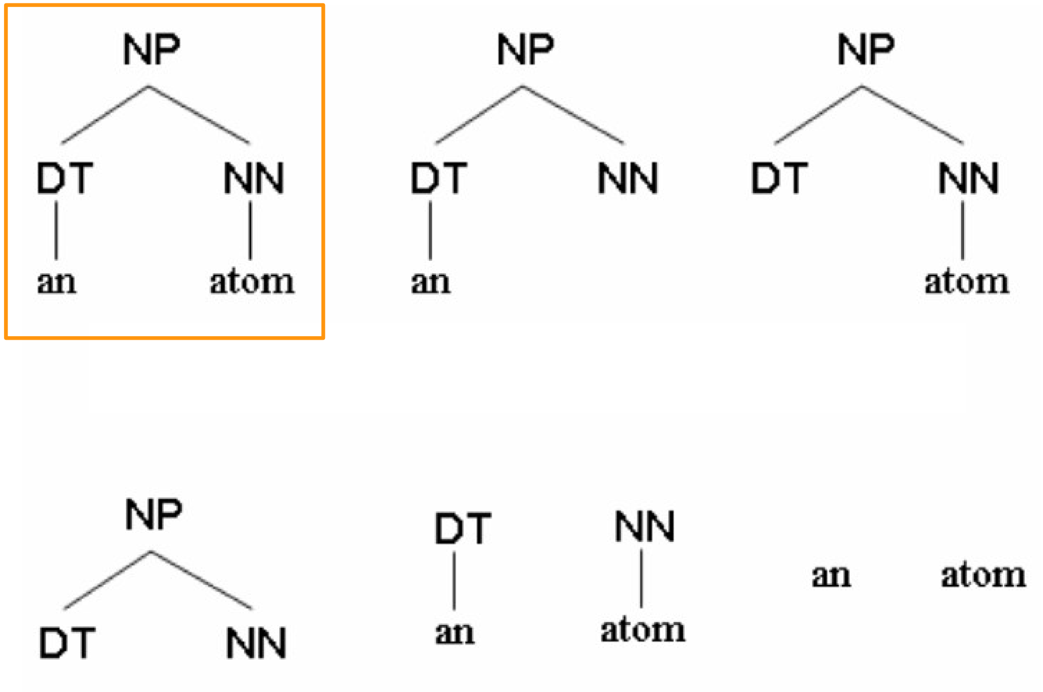
Tree Fragments



Tree fragment

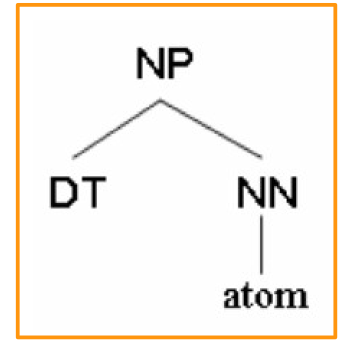
- at least one production rule / terminal symbol
- no incomplete production rule

Tree Fragments



Tree Fragments: Weight

tree fragment **i**

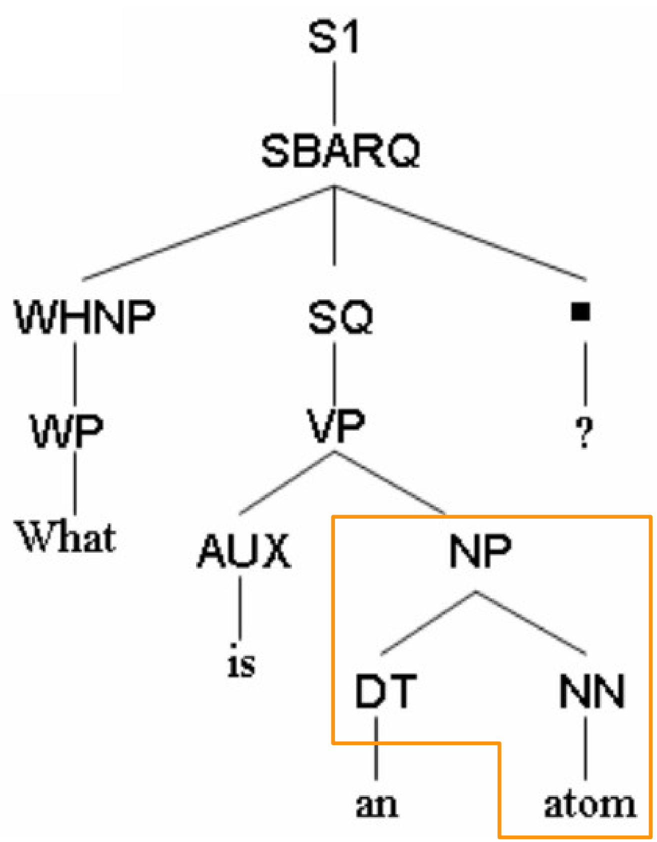


s(i) = size of **i** (here: 2)

$$v_i(T) = \begin{cases} \sqrt{\lambda}^{\,s(i)} \cdot \sqrt{\mu}^{\,d(i)} & \text{if } i \text{ is in } T \\ 0 & \text{otherwise} \end{cases}$$

$$0 \leq \lambda \leq 1,\ 0 \leq \mu \leq 1$$

d(i) = depth of **i** in **T**
(here: 4)



syntax tree **T**

Tree Kernel

$$\boldsymbol{v}(T) = \begin{bmatrix} v_1(T) \\ v_2(T) \\ \vdots \\ v_m(T) \end{bmatrix}$$ for all tree fragments v_i

$$K(T_1, T_2) = \boldsymbol{v}(T_1) \cdot \boldsymbol{v}(T_2)$$

dynamic programming algorithm in $O(|N_1| \cdot |N_2|)$

Summary and evaluation

- Approach 1: **Hiearchical Classifiers**
  - **Coarse-grained categories:** 91.00 % accuracy
  - **Fine-grained categories:** 84.20 % accuracy

- Approach 2: **Support Vector Machines with tree kernels**
  - **Coarse-grained categories:** 90.00 % accuracy
  - **Fine-grained categories:** Slight improvements compared to word/n-gram kernel
    (“*The experiment results are omitted to save space*”)

training set: 5500 questions, **test set:** 500 questions

References

- Xin Li, Dan Roth: *Learning Question Classifiers*, COLING Conference, 2002
- Dell Zhang, Wee Sun Lee: *Question Classification using Support Vector Machines*, SIGIR Conference, 2003
- Xin Li, Dan Roth: *Learning Question Classifiers: The Role of Semantic Information*, Journal of Natural Language Engineering, 2004
- Christopher J.C. Burges: *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining and Knowledge Discovery 2, 121-167, 1998
- Andrew Ng: *CS229 Lecture Notes Part V: Support Vector Machines*, Stanford Univerity