Advanced learning for Web Mining

MLxWM a.a. 2009/10

R. Basili

joint work with A. Moschitti,

Dipartimento di Informatica Sistemi e Produzione
Università di Roma “Tor Vergata”
Email: basili@info.uniroma2.it
Summary

- SVM and kernel based Learning
- Advanced kernels for complex ML tasks
- Information Extraction Applications
  - Semantic Role Labeling
  - Question Classification in Question Answering
  - Prestospace
    - Search through Video News Enrichment
Generative vs. Discriminative Learning for IE and NLP tasks

- **Generative models** (e.g. HMMs) require
  - The design of a model of *visible* and *hidden* variables
  - The definition of *laws of association* between hidden and visible variables
  - *Robust estimation methods* from the available samples

- **Limitations:**
  - Lack of precise generative models for language phenomena
  - Data sparseness: most of the language phenomena are simply too rare for robust estimation even in large samples
Generative vs. Discriminative Learning

- **Discriminative models** are not tight to any model (i.e. specific association among the problem variables).
- They learn to discriminate negative from positive evidence without building an explicit model of the target property.
- They derive useful evidence from training data only through observed individual features by optimizing some function of the recognition task (e.g. error).
- Examples of discriminative models are the perceptrons (i.e. linear classifiers).
Linear Classifiers (1)

An hyperplane has equation:

\[ f(\vec{x}) = \vec{x} \cdot \vec{w} + b, \quad \vec{x}, \vec{w} \in \mathbb{R}^n, b \in \mathbb{R} \]

\( \vec{x} \) is the vector of the instance to be classified
\( \vec{w} \) is the hyperplane gradient

Classification function:

\[ h(x) = \text{sign}(f(x)) \]
Linear Classifiers (2)

• Computationally simple.
• Basic idea: select an hypothesis that makes no mistake over training-set.
• The separating function is equivalent to a neural net with just one neuron (perceptron)
A neuron

- Synapse
- Soma
- Axon
- Dendrite
Perceptron

\[ \varphi(\bar{x}) = \text{sgn} \left( \sum_{i=1}^{n} w_i \times x_i + b \right) \]
Notations

• **The functional margin of an example** \((\vec{x}_i, y_i)\) wrt an hyperplane is: \(\gamma_i = y_i(\vec{w} \cdot \vec{x}_i + b)\)

• **The distribution of functional margins** of an hyperplane \((\vec{w}, b)\) wrt a training set \(S\) is the distribution of margins of all the examples in \(S\).

• **The functional margin of an hyperplane** is the minimum among the margins of the distribution
Notations 2

- By normalizing the hyperplane equation, i.e.
  $\left( \frac{\vec{w}}{\|\vec{w}\|}, \frac{b}{\|\vec{w}\|} \right)$ we get the geometric margin

- It measures the euclidean distance of points from the hyperplane

- The margin of a training set $S$ is the maximal geometric margin among all the hyperplanes.

- The corresponding hyperplane is called the maximal margin hyperplane
Just in case ...

• From $\cos(\vec{x}, \vec{w}) = \frac{\vec{x} \cdot \vec{w}}{\| \vec{x} \| \cdot \| \vec{w} \|}$

It follows that

$$\| \vec{x} \| \cos(\vec{x}, \vec{w}) = \frac{\vec{x} \cdot \vec{w}}{\| \vec{w} \|} = \vec{x} \cdot \frac{\vec{w}}{\| \vec{w} \|}$$
Geometric Margin
Geometric margin vs. data points in the training set
Maximal margin vs other margins
Perceptron: on-line algorithm

\[ \vec{w}_0 \leftarrow \vec{0}; b_0 \leftarrow 0; k \leftarrow 0; R \leftarrow \max_{1 \leq i \leq l} ||\vec{x}_i|| \]

Repeat

for \( i = 1 \) to \( \ell \)

if \( y_i (\vec{w}_k \cdot \vec{x}_i + b_k) \leq 0 \) then

\[ \vec{w}_{k+1} = \vec{w}_k + \eta y_i \vec{x}_i \]

\[ b_{k+1} = b_k + \eta y_i R^2 \]

\( k = k + 1 \)

endif

endfor

until no error is found

return \( k, (\vec{w}_k, b_k) \)
Duality

- The decision function of linear classifiers can be written as follows:
  \[ h(x) = \text{sgn}(\bar{w} \cdot \bar{x} + b) = \text{sgn}( (\sum_{j=1}^{\ell} \alpha_j y_j \bar{x}_j) \cdot \bar{x} + b) = \]
  \[ \text{sgn}( \sum_{i=1}^{\ell} \alpha_j y_j \bar{x}_j \cdot \bar{x} + b) \]

- as well the adjustment function
  \[ \text{if } y_i (\sum_{j=1}^{\ell} \alpha_j y_j \bar{x}_j \cdot \bar{x}_i + b) \leq 0 \text{ then } \alpha_i = \alpha_i + \eta \]

- The learning rate \( \eta \) impacts only in the re-scaling of the hyperplanes, and does not influence the algorithm (\( \eta = 1 \))

\( \Rightarrow \text{Training data only appear in the scalar products!!} \)
Which hyperplane?
Maximum Margin Hyperplanes

IDEA: Select the hyperplane that maximizes the margin
How to get the maximum margin?

The geometric margin is:

\[ \frac{2|k|}{\|w\|} \]

Optimization problem

\[ \text{MAX} \quad \frac{2|k|}{\|\bar{w}\|} \]

\[ \bar{w} \cdot \bar{x} + b \geq +k, \text{ if } \bar{x} \text{ is a positive ex.} \]

\[ \bar{w} \cdot \bar{x} + b \leq -k, \text{ if } \bar{x} \text{ is a negative ex.} \]
The optimization problem

- The optimal hyperplane satisfies:
  - Minimize \( \tau(\vec{w}) = \frac{1}{2} \|\vec{w}\|^2 \)
  - Under: \( y_i ((\vec{w} \cdot \vec{x}_i) + b) \geq 1, i = 1, \ldots, l \)

- The dual problem is simpler
The Lagrange Theory

Def. 2.24 Let \( f(\vec{w}) \), \( h_i(\vec{w}) \) and \( g_i(\vec{w}) \) be the objective function, the equality constraints and the inequality constraints (i.e. \( \geq \)) of an optimization problem, and let \( L(\vec{w}, \vec{\alpha}, \vec{\beta}) \) be its Lagrangian, defined as follows:

\[
L(\vec{w}, \vec{\alpha}, \vec{\beta}) = f(\vec{w}) + \sum_{i=1}^{m} \alpha_i g_i(\vec{w}) + \sum_{i=1}^{l} \beta_i h_i(\vec{w})
\]

\[
\tau(\vec{w}) = \frac{1}{2} \|\vec{w}\|^2
\]

\[
y_i ((\vec{w} \cdot \vec{x}_i) + b) \geq 1, i = 1,...,l
\]
Dual optimization problem

The Lagrangian dual problem of the above primal problem is

maximize \( \theta(\vec{\alpha}, \vec{\beta}) \)

subject to \( \vec{\alpha} \geq \vec{0} \)

where \( \theta(\vec{\alpha}, \vec{\beta}) = \inf_{w \in W} \ L(w, \vec{\alpha}, \vec{\beta}) \)
The dual formulation

The Lagrangian of the original problem becomes

$$L(\vec{w}, b, \vec{\alpha}) = \frac{1}{2} \vec{w} \cdot \vec{w} - \sum_{i=1}^{m} \alpha_i [y_i (\vec{w} \cdot \vec{x}_i + b) - 1]$$

- To solve it we must compute:

$$\theta(\vec{\alpha}, \vec{\beta}) = \inf_{w \in W} L(\vec{w}, \vec{\alpha}, \vec{\beta})$$

And impose derivatives = 0, wrt to $\vec{w}$

$$\frac{\partial L(\vec{w}, b, \vec{\alpha})}{\partial \vec{w}} = \vec{w} - \sum_{i=1}^{m} y_i \alpha_i \vec{x}_i = \vec{0} \quad \Rightarrow \quad \vec{w} = \sum_{i=1}^{m} y_i \alpha_i \vec{x}_i$$
The dual formulation (cont.)

- ... as well as wrt $b$

\[
\frac{\partial L(\tilde{w}, b, \alpha)}{\partial b} = \sum_{i=1}^{m} y_i \alpha_i = 0
\]

- The objective function also becomes:

\[
L(\tilde{w}, b, \alpha) = \frac{1}{2} \tilde{w} \cdot \tilde{w} - \sum_{i=1}^{m} \alpha_i [y_i(\tilde{w} \cdot \tilde{x}_i + b) - 1] =
\]

\[
= \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \tilde{x}_i \cdot \tilde{x}_j - \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \tilde{x}_i \cdot \tilde{x}_j + \sum_{i=1}^{m} \alpha_i
\]

\[
= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \tilde{x}_i \cdot \tilde{x}_j
\]
Final (dual) problem

\[
\text{maximize} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \vec{x}_i \cdot \vec{x}_j \\
\text{subject to} \quad \alpha_i \geq 0, \quad i = 1, \ldots, m \\
\sum_{i=1}^{m} y_i \alpha_i = 0
\]
Teorema di Khun-Tucker

- Necessary and sufficient conditions to have an optimal solution:

\[
\begin{align*}
\frac{\partial L(\vec{w}^*, \vec{\alpha}^*, \vec{\beta}^*)}{\partial \vec{w}} &= 0 \\
\frac{\partial L(\vec{w}^*, \vec{\alpha}^*, \vec{\beta}^*)}{\partial \vec{\beta}} &= 0 \\
\alpha_i^* g_i(\vec{w}^*) &= 0, \quad i = 1, \ldots, m \\
g_i(\vec{w}^*) &\leq 0, \quad i = 1, \ldots, m \\
\alpha_i^* &\geq 0, \quad i = 1, \ldots, m
\end{align*}
\]
Some consequences

- Lagrange constraints: \( \sum_{i=1}^{l} a_i y_i = 0 \quad \tilde{w} = \sum_{i=1}^{l} \alpha_i y_i \bar{x}_i \)
- Karush-Kuhn-Tucker constraints
  \( \alpha_i \cdot [y_i (\bar{x}_i \cdot \tilde{w} + b) - 1] = 0, \quad i = 1, \ldots, l \)
- The support vector are \( \bar{x}_i \) having not null \( \alpha_i \), i.e. such that \( y_i (\bar{x}_i \cdot \tilde{w} + b) = -1 \)
  They lie on the frontier
- \( b \) is derived through the following formula
  \[
  b^* = -\frac{\bar{w}^* \cdot \bar{x}^+ + \bar{w}^* \cdot \bar{x}^-}{2}
  \]
Support Vectors

Support vectors

Margin

Var₁

Var₂
Non linearly separable training data

Slack variables $\xi_i$ are introduced

Mistakes are allowed and optimization function is penalized
Soft Margin SVMs

New constraints:

\[ y_i (\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i \quad \forall \vec{x}_i \]
\[ \xi_i \geq 0 \]

Objective function:

\[ \min \frac{1}{2} ||\vec{w}||^2 + C \sum_i \xi_i \]

C is the trade-off between margin and errors
Dual optimization problem

\[
\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j (\vec{x}_i \cdot \vec{x}_j + \frac{1}{C} \delta_{ij})
\]

\[\alpha_i \geq 0, \quad \forall i = 1, \ldots, m\]

\[\sum_{i=1}^{m} y_i \alpha_i = 0\]
Soft Margin Support Vector Machines

\[
\min \frac{1}{2} \| \mathbf{w} \|^2 + C \sum \xi_i \quad \text{subject to } y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall \mathbf{x}_i \quad \xi_i \geq 0
\]

• The algorithm tries to keep \( \xi_i \) close to 0 and maximize the margin
• OBS: There is no attempt to minimize the number of errors (This is an NP-complete problem); The cumulative distance from the hyperplanes is minimized
• When \( C \to \infty \), the solution is the same as in the hard-margin setting
• \textit{Warning!!!}: when \( C = 0 \) we get \( \| \mathbf{w} \| = 0 \). As it is always true that \( y_i b \geq 1 - \xi_i \quad \forall \mathbf{x}_i \)
• When \( C \) is increased we tend to limit the number of errors.
Robustness: *Soft* vs *Hard Margin* SVMs

**Soft Margin SVM**

\[
\|w\| \cdot x + b = 0
\]

**Hard Margin SVM**

\[
\tilde{w} \cdot \tilde{x} + b = 0
\]
Soft vs Hard Margin SVMs

- A Soft-Margin SVM has always a solution
- A Soft-Margin SVM is more robust wrt odd training examples
  - Insufficient Vocabularies
  - High ambiguity of (linguistic) features
- An Hard-Margin SVM requires no parameter
SVM: References


• A tutorial on Support Vector Machines for Pattern Recognition (C.J.Burges)

• The Vapnik-Chervonenkis Dimension and the Learning Capability of Neural Nets (E.D: Sontag)

• Computational Learning Theory
  (Sally A Goldman Washington University St. Louis Missouri)
  – http://www.learningtheory.org/articles/COLTSurveyArticle.ps

• AN INTRODUCTION TO SUPPORT VECTOR MACHINES (and other kernel-based learning methods), N. Cristianini and J. Shawe-Taylor Cambridge University Press.

• The Nature of Statistical Learning Theory, V. N. Vapnik - Springer Verlag (December, 1999)
SVM-light: an SVM tool

- It implements the *soft margin approach*
- Make available the procedures to solve the optimization problem
- It supports a binary classification problem
- Examples and specifications at:
  http://www.joachims.org/
  (http://svmlight.joachims.org/)
Kernel-based Learning: Outline

• Metodi Kernel
  – Motivazioni
  – Esempio

• Kernel standard
  – Polynomial kernel
  – String Kernel

• Kernel *avanzati*
  – Lexical Kernel
  – Tree kernel
La funzione Kernel

• Il learning dipende solo dal prodotto scalare dei vettori di esempio
• Quindi dipende dalla Gram-matrix,
• In generale si definisce kernel la funzione:
\[ K(\tilde{z}, \tilde{x}) = \phi(\tilde{z}) \cdot \phi(\tilde{x}) \]
• La funzione kernel produce il risultato a partire dagli oggetti iniziali
• Quando la mappatura \( \phi \) è l’identità abbiamo l’usuale prodotto scalare.
• Gli esempi compaiono nell’algoritmo di learning (ad es. il percettrone) solo attraverso i loro contributi al kernel
Primo Vantaggio: rendere linearmente separabili gli esempi

- Mappare i dati in uno Spazio di Feature dove sono linearmente separabili $\bar{x} \rightarrow \phi(\bar{x})$
  (i.e. attributi $\rightarrow$ feature)
Esempio di una funzione di mappatura

- Due masse $m_1$ e $m_2$, una vincolata
- Applico una forza $f_a$ alla massa $m_1$
- Esperimenti
  - Features $m_1$, $m_2$ e $f_a$
- Supponiamo di volere apprendere quando $m_1$ si allontana da $m_2$
- Considerando la legge gravitazionale di Newton

$$f(m_1, m_2, r) = C \frac{m_1 m_2}{r^2}$$
- Dobbiamo trovare quando $f(m_1, m_2, r) < f_a$
Esempio di una funzione di mappatura

\[ \vec{x} = (x_1, \ldots, x_n) \rightarrow \phi(\vec{x}) = (\phi_1(\vec{x}), \ldots, \phi_n(\vec{x})) \]

- Non esprimibile linearmente, quindi cambio spazio

\[(f_a, m_1, m_2, r) \rightarrow (k, x, y, z) = (\ln f_a, \ln m_1, \ln m_2, \ln r)\]

- Poiché

\[ \ln f(m_1, m_2, r) = \ln C + \ln m_1 + \ln m_2 - 2\ln r = c + x + y - 2z \]

- Allora l'iperpiano è la funzione richiesta

\[ \ln f_a - \ln m_1 - \ln m_2 + 2\ln r - \ln C = 0 \]

\[(\ln m_1, \ln m_2, -2\ln r) \cdot (x, y, z) - \ln f_a + \ln C = 0, \text{ posso decidere senza errore se le masse si avvicinano o si allontanano}\]
Feature Spaces and Kernels

• Feature Space
  • Lo spazio di input è mappato in un nuovo spazio dotato di prodotto scalare \( F \) (detto *feature space*) attraverso una trasformazione (non lineare)
    \[
    \phi = R^N \rightarrow F
    \]

• Kernel
  • La valutazione della funzione di decisione richiede il prodotto scalare ma mai i pattern rimappati in forma esplicita \( \phi(x) \)
  • Il prodotto scalare viene calcolato attraverso la funzione kernel
    \[
    k(x, y) = (\phi(x) \cdot \phi(y))
    \]
Funzione di classificazione: forma duale

\[
\text{sgn}(\mathbf{w} \cdot \mathbf{x} + b) = \text{sgn}\left( \sum_{j=1}^{\ell} \alpha_j y_j \mathbf{x}_j \cdot \mathbf{x} + b \right)
\]

- Notare che i dati appaiono solo nel prodotto scalare.
- La matrice \( G = \left( \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \right)_{i,j=1}^{\ell} \) è chiamata Gram matrix.
Possiamo riscrivere la funzione di decisione:

\[ h(x) = \text{sgn}(\mathbf{w} \cdot \phi(\mathbf{x}) + b) = \text{sgn}(\sum_{j=1}^{\ell} \alpha_j y_j \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}) + b) = \text{sgn}(\sum_{i=1}^{\ell} \alpha_j y_j k(\mathbf{x}_j, \mathbf{x}) + b) \]

Così come la funzione di aggiornamento

\[
\text{if } y_i (\sum_{j=1}^{\ell} \alpha_j y_j \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_i) + b = \sum_{j=1}^{\ell} \alpha_j y_j k(\mathbf{x}_j, \mathbf{x}_i) + b) \leq 0
\]

\[ \text{then } \alpha_i = \alpha_i + \eta \]

Il learning rate \( \eta \) non influenza l’algoritmo (teorema Novikoff) quindi fissiamo \( \eta = 1 \).
Kernels in Support Vector Machines

• Nel problema Soft Margin SVMs si deve massimizzare:

\[
\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \vec{x}_i \cdot \vec{x}_j + \frac{1}{2C} \vec{\alpha} \cdot \vec{\alpha} - \frac{1}{C} \vec{\alpha} \cdot \vec{\alpha}
\]

Usando le funzioni kernel possiamo riscrivere il problema come:

\[
\begin{align*}
\text{maximize} & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j (k(o_i, o_j) + \frac{1}{C} \delta_{ij}) \\
\alpha_i & \geq 0, \quad \forall i = 1, \ldots, m \\
\sum_{i=1}^{m} y_i \alpha_i & = 0
\end{align*}
\]
Definizione delle Funzioni Kernel

**Def. 2.26** A kernel is a function $k$, such that $\forall \bar{x}, \bar{z} \in X$

$$k(\bar{x}, \bar{z}) = \phi(\bar{x}) \cdot \phi(\bar{z})$$

where $\phi$ is a mapping from $X$ to an (inner product) feature space.

- Le funzioni kernel esprimono mappature implicite di questo tipo

$$\bar{x} \in \mathbb{R}^n, \quad \hat{\phi}(\bar{x}) = (\phi_1(\bar{x}), \phi_2(\bar{x}), \ldots, \phi_m(\bar{x})) \in \mathbb{R}^m$$
Validità delle funzioni kernel (1)

Def. B.11  Eigen Values
Given a matrix $A \in \mathbb{R}^m \times \mathbb{R}^n$, an egeinvalue $\lambda$ and an egeinvector $\vec{x} \in \mathbb{R}^n - \{\vec{0}\}$ are such that

$$A\vec{x} = \lambda\vec{x}$$

Def. B.12  Symmetric Matrix
A square matrix $A \in \mathbb{R}^n \times \mathbb{R}^n$ is symmetric iff $A_{ij} = A_{ji}$ for $i \neq j$ $i = 1, .., m$ and $j = 1, .., n$, i.e. iff $A = A'$.

Def. B.13  Positive (Semi-) definite Matrix
A square matrix $A \in \mathbb{R}^n \times \mathbb{R}^n$ is said to be positive (semi-) definite if its eigenvalues are all positive (non-negative).
L’idea principale di questa proposizione è che se la Gram matrix è semidefinita positiva allora esiste il mapping \( \phi \) che realizza la funzione kernel, cioè uno spazio \( F \) in cui la separabilità è espressa in modo migliore.
Feature Spaces and Kernels

• Esempio di Kernel
  - Polynomial kernel
    \[ k(x, y) = (x \cdot y)^d \]
  
• Se \( d=2 \) e \( x, y \in \mathbb{R}^2 \)

\[
(x \cdot y)^2 = \left( \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right)^2 = \left( \begin{bmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{bmatrix} \cdot \begin{bmatrix} y_1^2 \\ \sqrt{2} y_1 y_2 \\ y_2^2 \end{bmatrix} \right) = (\phi(x) \cdot \phi(y)) = k(x, y)
\]
Polynomial Kernel ($n$ dimensions)

\[
(\vec{x} \cdot \vec{z})^2 = \left( \sum_{i=1}^{n} x_i z_i \right)^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j z_i z_j = \sum_{k=1}^{m} X_k Z_k
\]

\[
= \left( \sum_{i=1}^{n} x_i z_i \right) \left( \sum_{j=1}^{n} x_i z_i \right) = \sum_{i, j \in \{1, \ldots, n\}} (x_i x_j)(z_i z_j) = \vec{X} \cdot \vec{Z}
\]
General Polynomial Kernel \((n\ \text{dimensions})\)

\[
(\vec{x} \cdot \vec{z} + c)^2 = \left( \sum_{i=1}^{n} x_i z_i + c \right)^2 = \left( \sum_{i=1}^{n} x_i z_i + c \right) \left( \sum_{j=1}^{n} x_j z_j + c \right) = \\
= \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j z_i z_j + 2c \sum_{i=1}^{n} x_i z_i + c^2 = \\
= \sum_{i,j \in \{1, \ldots, n\}} (x_i x_j)(z_i z_j) + \sum_{i=1}^{n} (\sqrt{2cx_i})(\sqrt{2cz_i}) + c^2
\]
Polynomial kernel and the conjunction of features

- The initial vectors can be mapped into a higher dimensional space \( c=1 \)
  \[
  \Phi(<x_1, x_2>) \rightarrow (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)
  \]

- More expressive, as \( (x_1x_2) \) encodes \( stock+market \) vs. \( downtown+market \) features

- We can smartly compute the scalar product as
  \[
  \Phi(\bar{x}) \cdot \Phi(\bar{z}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1) \cdot (z_1^2, z_2^2, \sqrt{2}z_1z_2, \sqrt{2}z_1, \sqrt{2}z_2, 1) = \\
  = x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 + 2x_1z_1 + 2x_2z_2 + 1 = \\
  = (x_1z_1 + x_2z_2 + 1)^2 = (\bar{x} \cdot \bar{z} + 1)^2 = K_{p^2}(\bar{x}, \bar{z})
  \]
Architettura di una SVM

- Classificatore non lineare (basato su kernel)

  - La funzione di decisione è
    \[ f(x) = \text{sgn}\left(\sum_{i=1}^{l} v_i \phi(x) \cdot \phi(x_i) + b\right) \]
    
    \[ = \text{sgn}\left(\sum_{i=1}^{l} v_i k(x, x_i) + b\right) \]

  \[ \phi(x_i) \] sostituisce ogni esempio di training \( x_i \)

  \[ v_i = \alpha_i y_i \]

  \( v_i \) sono calcolate attraverso la soluzione del problema di ottimizzazione
String Kernel

- Given two strings, the number of matches between their substrings is computed
- E.g. Bank and Rank
  - B, a, n, k, Ba, Ban, Bank, an, ank, nk
  - R, a, n, k, Ra, Ran, Rank, an, ank, nk
- String kernel over sentences and texts
- Huge space but there are efficient algorithms
Formal Definition

\[ s = s_1, \ldots, s_{|s|} \]

\[ \vec{I} = (i_1, \ldots, i_{|u|}) \quad u = s[\vec{I}], \text{ substring of } s \text{ defined by } \vec{I} \]

\[ \phi_u(s) = \sum_{\vec{I} : u = s[\vec{I}]} \lambda^{|l(\vec{I})|} \text{, } \text{con} \quad l(\vec{I}) = i_{|u|} - i_1 + 1 \]

\[ K(s, t) = \sum_{u \in \Sigma^*} \phi_u(s) \cdot \phi_u(t) = \sum_{u \in \Sigma^*} \sum_{\vec{I} : u = s[\vec{I}]} \lambda^{|l(\vec{I})|} \sum_{\vec{J} : u = t[\vec{J}]} \lambda^{|l(\vec{J})|} = \]

\[ = \sum_{u \in \Sigma^*} \sum_{\vec{I} : u = s[\vec{I}]} \sum_{\vec{J} : u = t[\vec{J}]} \lambda^{|l(\vec{I})| + |l(\vec{J})|} \text{, con } \Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n \]
Kernel tra **Bank** e **Rank**

B, a, n, k, Ba, Ban, Bank, an, ank, nk, Bn, Bnk, Bk and ak are the substrings of **Bank**.

R, a, n, k, Ra, Ran, Rank, an, ank, nk, Rn, Rnk, Rk and ak are the substrings of **Rank**.

- **Common substrings:**
  - a, n, k, an, ank, nk
An example of string kernel computation

- $\phi_a(\text{Bank}) = \phi_a(\text{Rank}) = \lambda^{(i_1 - i_1 + 1)} = \lambda^{(2 - 2 + 1)} = \lambda$,
- $\phi_n(\text{Bank}) = \phi_n(\text{Rank}) = \lambda^{(i_1 - i_1 + 1)} = \lambda^{(3 - 3 + 1)} = \lambda$,
- $\phi_k(\text{Bank}) = \phi_k(\text{Rank}) = \lambda^{(i_1 - i_1 + 1)} = \lambda^{(4 - 4 + 1)} = \lambda$,
- $\phi_{an}(\text{Bank}) = \phi_{an}(\text{Rank}) = \lambda^{(i_1 - i_2 + 1)} = \lambda^{(3 - 2 + 1)} = \lambda^2$,
- $\phi_{ank}(\text{Bank}) = \phi_{ank}(\text{Rank}) = \lambda^{(i_1 - i_3 + 1)} = \lambda^{(4 - 2 + 1)} = \lambda^3$,

$\phi_{nk}(\text{Bank}) = \phi_{nk}(\text{Rank}) = \lambda^{(i_1 - i_2 + 1)} = \lambda^{(4 - 3 + 1)} = \lambda^2$,

$\phi_{ak}(\text{Bank}) = \phi_{ak}(\text{Rank}) = \lambda^{(i_1 - i_2 + 1)} = \lambda^{(4 - 2 + 1)} = \lambda^3$.

It follows that $K(\text{Bank, Rank}) = (\lambda, \lambda, \lambda, \lambda^2, \lambda^3, \lambda^2, \lambda^3) \cdot (\lambda, \lambda, \lambda, \lambda^2, \lambda^3, \lambda^2, \lambda^3) = 3\lambda^2 + 2\lambda^4 + 2\lambda^6.$
Kernel Lessicale Semantico

- I sistemi di Text Classification agiscono sulle rappresentazioni vettoriali $\tilde{d}$ dei documenti $d$
- Dato un documento $d$ posso tentare di rappresentarlo in uno spazio dei concetti (cioè' sensi secondo Wordnet) invece che nello spazio di parole (VSM tradizionale)
- Questo genera una immagine (complessa) dei sensi delle parole di $d$ che chiameremo $\Phi(d)$
- Per apprendere da esempi $d_1$ e $d_2$ si può utilizzare una funzione della similitudine tra tutti i termini di $d_1$ e quelli di $d_2$, con un kernel (semantico lessicale) che calcola quindi $SK(d_1, d_2) = \Phi(d_1) \cdot \Phi(d_2)$
Kernel Lessicale Semantico (2)

• La similarità tra due documenti, $d_1$ e $d_2$, è data dal kernel semantico SK seguente:

$$SK(d_1, d_2) = \sum_{w_1 \in d_1, w_2 \in d_2} \sigma(w_1, w_2)$$

• dove $\sigma$ è una funzione della similarità semantica in Wordnet tra due parole

• Una possibile scelta di $\sigma$ è data dalla conceptual density in Wordnet
OSS: in uno VSM tradizionale la similarità tra i due documenti sarebbe 0 perché non ci sono parole comuni tra Doc1 e Doc2!!
The effects of WN in poor training conditions

<table>
<thead>
<tr>
<th>Category</th>
<th>24 docs</th>
<th></th>
<th>160 docs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bow</td>
<td>SK</td>
<td>bow</td>
<td>SK</td>
</tr>
<tr>
<td>Acq.</td>
<td>55.3±18.1</td>
<td>50.8±18.1</td>
<td>86.7±4.6</td>
<td>84.2±4.3</td>
</tr>
<tr>
<td>Crude</td>
<td>3.4±5.6</td>
<td>3.5±5.7</td>
<td>64.0±20.6</td>
<td>62.0±16.7</td>
</tr>
<tr>
<td>Earn</td>
<td>64.0±10.0</td>
<td>64.7±10.3</td>
<td>91.3±5.5</td>
<td>90.4±5.1</td>
</tr>
<tr>
<td>Grain</td>
<td>45.0±33.4</td>
<td>44.4±29.6</td>
<td>69.9±16.3</td>
<td>73.7±14.8</td>
</tr>
<tr>
<td>Interest</td>
<td>23.9±29.9</td>
<td>24.9±28.6</td>
<td>67.2±12.9</td>
<td>59.8±12.6</td>
</tr>
<tr>
<td>Money-fx</td>
<td>36.1±34.3</td>
<td>39.2±29.5</td>
<td>69.1±11.9</td>
<td>67.4±13.3</td>
</tr>
<tr>
<td>Trade</td>
<td>9.8±21.2</td>
<td>10.3±17.9</td>
<td>57.1±23.8</td>
<td>60.1±15.4</td>
</tr>
<tr>
<td>Wheat</td>
<td>8.6±19.7</td>
<td>13.3±26.3</td>
<td>23.9±24.8</td>
<td>31.2±23.0</td>
</tr>
<tr>
<td>Mic.Avg.</td>
<td>37.2±5.9</td>
<td>41.7±6.0</td>
<td>75.9±11.0</td>
<td>77.9±5.7</td>
</tr>
</tbody>
</table>

Table 2: Performance of the linear and Semantic Kernel with 40 and 160 training documents over 8 categories of the Reuters
Figure 1: MicroAverage $F_1$ of SVMs using bow, SK and SK-POS kernels over the 8 categories of 20NewsGroups.
Tree kernels: Motivations

• Classification over highly structured data require complex similarity functions

• Modeling syntax in Natural Language learning tasks is crucial, e.g.
  – Syntactic disambiguation
  – Semantic role relations within predicate argument structures
  – Question Classification

• Tree kernels are natural way to exploit syntactic information from sentence parse trees
  – useful to engineer novel and complex features.

• How do different tree kernels impact on different parsing paradigms and different tasks?

• Are they efficient in practical applications?
The Collins and Duffy’s Tree Kernel (called SST in [Vishwanathan and Smola, 2002])

```
VP
  V
  NP
  gives
  D
  N
  a
  talk
```
The overall fragment set
Making Explicit the Feature Space

\[ \vec{x} = (0, \ldots, 1, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 0, \ldots, 1, \ldots, 0) \]

- \( \vec{x}_1 \cdot \vec{x}_2 \) counts the number of common substructures
Computing over the implicit representation

\[ \bar{x}_1 \cdot \bar{x}_2 = \phi(T_1) \cdot \phi(T_2) = K(T_1, T_2) = \]
\[ = \sum_{n_1 \in T_1} \sum_{n_2 \in T_2} \Delta(n_1, n_2) \]
Implicit Representation

\[ \bar{x}_1 \cdot \bar{x}_2 = \phi(T_1) \cdot \phi(T_2) = K(T_1, T_2) = \sum_{n_1 \in T_1} \sum_{n_2 \in T_2} \Delta(n_1, n_2) \]

- [Collins and Duffy, ACL 2002] evaluate \( \Delta \) in \( O(n^2) \):

\[
\Delta(n_1, n_2) = \begin{cases} 
0, & \text{if the productions are different} \\
1, & \text{if pre-terminals} \\
\prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j))) & \text{else}
\end{cases}
\]
Weighting

- Decay factor
  \[ \Delta(n_1, n_2) = \lambda, \quad \text{if pre-terminals else} \]
  \[ \Delta(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j))) \]

- Normalization
  \[ K'(T_1, T_2) = \frac{K(T_1, T_2)}{\sqrt{K(T_1, T_1) \times K(T_2, T_2)}} \]
SubTree (ST) Kernel [Vishwanathan and Smola, 2002]
ST Kernel Evaluation

• Given the equation for the SST kernel (subset), i.e.

\[
\Delta(n_1, n_1) = 1, \text{ if pre-terminals else } \\
\Delta(n_1, n_2) = 0, \text{ if the productions are different else } \\
\Delta(n_1, n_2) = \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j)))
\]

• ...
Evaluation

• Given the equation for the ST kernel

\[ \Delta(n_1, n_2) = 0, \text{ if the productions are different}\]
\[ \Delta(n_1, n_2) = 1, \text{ if pre-terminals else}\]

\[ \Delta(n_1, n_2) = \prod_{j=1}^{nc(n_1)} (\Delta(ch(n_1, j), ch(n_2, j))) \]
Labeled Ordered Tree Kernel

- SST satisfies the constraint “remove 0 or all children at a time”.
- If we relax such constraint we get more general substructures [Kashima and Koyanagi, 2002]
Natural Language Processing Applications

• We have different kernels that induce different feature spaces.
• How should such kernel functions be used?
• An answer can be given to the problem of encoding syntactic information.
• As example we study two different tasks requiring syntactic information.
Semantic Role Labeling

• Given an event:
  – Some words describe the relation among different participants
  – Such words can be considered predicates
  – The participants are their arguments.

• Example:
  Paul gives a lecture in Rome
Semantic Role Labeling

• Given an event:
  – Some words describe the relation among different participants
  – Such words can be considered predicates
  – The participants are their arguments.

• Example:

  \[
  [\text{Arg}_0 \text{ Paul}] [\text{predicate} \text{ gives} [\text{Arg}_1 \text{ a lecture}] [\text{Arg}_M \text{ in Rome}]
  \]
Semantic Role Labeling

• Given an event:
  – Some words describe the relation among different participants
  – Such words can be considered predicates
  – The participants are their arguments.

• Example:

  \[ [\text{Argo Paul}] [\text{predicate gives} [\text{Arg1 a lecture}] [\text{ArgM in Rome}]] \]

• PropBank and FrameNet propose two different theories and resources
Semantic/Syntactic structures

- **Given a sentence with its semantic annotation:**

\[
[S_{Arg0 \text{ Paul}}] [_{\text{predicate}} \text{ gives} [S_{Arg1 \text{ a lecture}}] [S_{ArgM \text{ in Rome}}]]
\]

```
  S
 /\  \
|  |  |
NP VP
 /\  \
|  |  |
N VP NP PP
 /\  /\  /\  \
Paul | gives | a | in \\
     |   | lecture | Rome
     |   |         |     \\
     Arg. 0 | Predicate | Arg. 1 | Arg. M
```
A Tree Kernel for Semantic Role labeling

Paul gives a talk in a formal style

Arg. 1
Linking syntax to semantics

• Police arrested the man for shoplifting
Semantics in NLP: Resources

• Lexicalized Models
  – Propbank
  – NomBank

• Framenet
  – Inspired by frame semantics
  – Frames are lexicalized prototyopes for real-world situations
  – Participants are called frame elements (roles)
PropBank (Palmer et al., 2005)

- Transfer sentences to propositions
  - \textit{Kristina hit Scott} $\Rightarrow$ \textit{hit(Kristina,Scott)}
- Penn TreeBank $\Rightarrow$ PropBank
- Add a semantic layer on Penn TreeBank
  - Define a set of semantic roles for each verb
  - Each verb’s roles are numbered.

- \textit{[A0/the company] to ... offer[A1/a 15% to 20% stake] [A2/to the public]}.
- \textit{[A0/Sotheby's] offered[A2/the Dorrance heirs] [A1/a money-back guarantee]}.
- \textit{[A1/an amendment] offered[A0/by Rep. Peter DeFazio]}..
- \textit{[A2/Subcontractors] will be offered[A1/a settlement]}.
PropBank – Frame files

- **hit.01 “strike”**
  - A0: agent, hitter; A1: thing hit; A2: instrument, thing hit by or with
  - \( [A_0 \text{Kristina}] \text{hit} [A_1 \text{Scott}] [A_2 \text{with a baseball}] \text{yesterday}. \)

- **look.02 “seeming”**
  - A0: seemer; A1: seemed like; A2: seemed to
  - \( [A_0 \text{It}] \text{looked} [A_2 \text{to her}] \text{like} [A_1 \text{he deserved this}]. \)

- **deserve.01 “deserve”**
  - A0: deserving entity; A1: thing deserved; A2: in-exchange-for
  - \( \text{It looked to her like } [A_0 \text{he}] \text{deserved} [A_1 \text{this}]. \)
PropBank – Data

- Current release (Mar 4, 2005)
- **Proposition Bank I**
  - Verb Lexicon: 3,324 frame files
  - Annotation: ~113,000 propositions
  - [http://www.cis.upenn.edu/~mpalmer/project_pages/ACE.htm](http://www.cis.upenn.edu/~mpalmer/project_pages/ACE.htm)
- Alternative format: CoNLL-04,05 shared task
  - Represented in table format
  - Has been used as standard data set for the shared tasks on semantic role labeling
  - [http://www.lsi.upc.es/~srlconll/soft.html](http://www.lsi.upc.es/~srlconll/soft.html)
Frame Semantics

• Research in Empirical Semantics suggests that words represents categories of experience (*situations*)

• A *frame* is a cognitive structuring device (i.e. a kind of prototype) indexed by *words* and used to support understanding (Fillmore, 1975)
  – *Lexical Units* evoke a Frame in a sentence

• Frames are made of *elements* that express participants to the situation (*Frame Elements*)

• During communication LUs evoke the frames
# Frame Semantics

## Frame: Killing

A Killer or Cause causes the death of the Victim.

<table>
<thead>
<tr>
<th>Frame Elements</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Killer</strong></td>
<td>John drowned Martha.</td>
</tr>
<tr>
<td><strong>Victim</strong></td>
<td>John drowned Martha.</td>
</tr>
<tr>
<td><strong>Means</strong></td>
<td>The flood <strong>exterminated</strong> the rats <strong>by cutting off access to food</strong>.</td>
</tr>
<tr>
<td><strong>Cause</strong></td>
<td><strong>The rockslide</strong> <strong>killed</strong> nearly half of the climbers.</td>
</tr>
<tr>
<td><strong>Instrument</strong></td>
<td>It’s difficult to <strong>suicide</strong> with only a <strong>pocketknife</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>annihilate.v, annihilation.n, asphyxiate.v, assassin.n, assassinate.v, assassination.n, behead.v, beheading.n, blood-bath.n, butcher.v, butchery.n, carnage.n, crucifixion.n, crucify.v, deadly.a, decapitate.v, decapitation.n, destroy.v, dispatch.v, drown.v, eliminate.v, euthanasia.n, euthanize.v, ...</td>
</tr>
</tbody>
</table>
Frame Semantics

• Lexical descriptions are expected to define the indexed frame and the frame elements with their realization at the syntactic level:
  – *John bought a computer from Janice for 1000 $*

• Mapping into syntactic arguments
  – the *buyer* is (usually) in the subject position

• Obligatory vs. optional arguments

• Selectional preferences
  – *The seller* and *the buyer* are usually “humans” or “social groups”
The FrameNet project

• The aims
  - Create a lexical resource by describing a significant portion of English in terms of precise and rich frame semantics

• The output
  - Frame Database: a structured system of Frames and Fes
  - Lexical database: syntactic and semantic descriptions of frame-evoking words (N,V,A)
  - Annotated Corpus: wide coverage examples
Committing Crime

Definition:

A **Perpetrator** (generally intentionally) commits a **Crime**, i.e. does something not permitted by the laws of society.

They **PERPETRATED** a felony by substituting a lie for negotiations.

The suspect had allegedly **COMMITTED** the crime to gain the attention of a female celebrity.

FEs:

Core:

**Crime** [Cr]

An act, generally intentional, that has been formally forbidden by law.

How can he **COMMIT** treason against the King of England in a foreign country, if he is not English?

He **PERPETRATED** a crime against mother nature.

**Perpetrator** [Perp]

The individual that commits a **Crime**.

How can he **COMMIT** treason against the King of England in a foreign country, if he is not English?

He **PERPETRATED** a crime against mother nature.

Non-Core:

**Frequency** [Freg]

The frequency with which a **crime** is committed.

The average serial killer **COMMITS** a crime **every five years**.

**Instrument** [Inst]

The **Instrument** used in committing the crime.

Most crimes are **COMMITTED** with a firearm.
The FrameNet Hierarchy

Reciprocity

6 children total

Commercial_transaction

Transfer

Commerce_goods-transfer

Commerce_money-transfer

Commerce_buy

Commerce_sell

Commerce_collect

Commerce_pay
Framenet - Data

• Methodology of constructing FrameNet
  - Define/discover/describe frames
  - Decide the participants (frame elements)
  - List lexical units that evoke the frame
  - Find example sentences in the BNC and annotate them

• Corpora
  - FrameNet I -British National Corpus only
  - FrameNet II -LDC North American Newswire corpora

• Size
  - >10,000 lexical units, >825 frames, >135,000 sentences

• http://framenet.icsi.berkeley.edu
Gold Standard Tree Experiments

- PropBank and PennTree bank
  - about 53,700 sentences
  - Sections from 2 to 21 train., 23 test., 1 and 22 dev.
  - Arguments from Argo to Arg5, ArgA and ArgM for
    a total of 122,774 and 7,359

- FrameNet experiments
SVM-light-TK Software

- Encodes ST, SST and PT in SVM-light [Joachims, 1999]
- Available at http://ai-nlp.info.uniroma2.it/moschitti/
- New extensions: tree forests, vector sets and the PT kernel coming soon
Semantic Role Labeling @ UTV

• An important application of SVM is Semantic Role labeling wrt Propbank or 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
SRL Demo

- Kernel-based system for SRL over raw texts
- based on the Charniak parser
- Adopts the Propbank standard but as also been applied to Framenet
Kernel-based Semantic Role Labeling

SRL User Interface

Enter a new sentence:

Select an example sentence:

- Couch-potato jocks watching ABC's 'Monday Night Football' can now vote during halftime for the greatest play in 20 years from among four or five filmed replays.

- John went to the supermarket on Friday and bought nine apples.

- My talk was intended to help you to understand this difficult topic.

- Jimmy's wife was looking sadly at her broken glasses.

- During last summer, two thousand trees were burnt by criminals.

- Mary would like to understand why John betrayed her.
Couch potato jocks are watching ABC’s “Monday Night Football” and now vote during halftime for the greatest play in 20 years from among four or five filmed replays.
Couch-potato jocks watching ABC's "Monday Night Football" can now vote during halftime for the greatest play in 20 years from among four or five filmed replays.
Capello will be officially unveiled on Monday and Leonardo believes that he is the right man to take England forward.
Capello will be officially unveiled on Monday and Leonardo believes that he is the right man to take England forward.
Capello will be officially unveiled on Monday and Leonardo believes that he is the right man to take England forward.
Automatic Predicate Argument Extraction

• Boundary Detection
  – One binary classifier

• Argument Type Classification
  – Multi-classification problem
  – $n$ binary classifiers (ONE-vs-ALL)
  – Select the argument with maximum score
Typical standard linear features
(Gildea & Jurafsky, 2002)

- Phrase Type of the argument
- Parse Tree Path, between the predicate and the argument
- Head word
- Predicate Word
- Position
- Voice
Features for the linear kernel in SRL

<table>
<thead>
<tr>
<th>Features</th>
<th>Argument identification</th>
<th>Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>TARGETLEMMMA</td>
<td>C, D</td>
<td></td>
</tr>
<tr>
<td>FES</td>
<td>C, D</td>
<td></td>
</tr>
<tr>
<td>TARGETPOS</td>
<td>C, D</td>
<td></td>
</tr>
<tr>
<td>VOICE</td>
<td>C, D</td>
<td></td>
</tr>
<tr>
<td>POSITION</td>
<td>C, D</td>
<td></td>
</tr>
<tr>
<td>ARGWORD/POS</td>
<td>C, D</td>
<td></td>
</tr>
<tr>
<td>LEFTWORD/POS</td>
<td>C, D</td>
<td></td>
</tr>
<tr>
<td>RIGHTWORD/POS</td>
<td>C, D</td>
<td></td>
</tr>
<tr>
<td>PARENTWORD/POS</td>
<td>C, D</td>
<td></td>
</tr>
<tr>
<td>C-SUBCAT</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>C-PATH</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>PHRASETYPE</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>GOVCAT</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>D-SUBCAT</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>D-PATH</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>CHILDDEPSET</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>PARENTHASOBJ</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>RELTOPARENT</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>FUNCTION</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C-SUBCAT. Subcategorization frame: corresponds to the phrase-structure rule used to expand the phrase around the target. For *give* in the example, this feature is $\text{VP} \rightarrow \text{VB} \ 	ext{NP} \ 	ext{NP}$.

C-PATH. A string representation of the path through the constituent tree from the target word to the argument constituent. For instance, the path from *gave* to *she* is $\uparrow \text{VP} \downarrow \uparrow \text{S-NP}$.

PHRASETYPE. Phrase type of the argument constituent, e.g. NP for *she*.

GOVCAT. Governing category: this feature is either S or VP, and is found by starting at the argument constituent and moving upwards until either a VP or a sentence node (S, SINV, or SQ) is found. For instance, for *she*, this feature is S, while for *the horse*, it is VP. This can be thought of as a very primitive way of distinguishing subjects and objects.

Table 1: Classifier features. The features that are part of the constituent-based and the dependency-based systems are marked C and D, respectively.
An example

Phrase Type

Predicate

Word

Head Word

Parse Tree

Path

Position Right

Voice Active

Predicate

S

N

NP

D

N

VP

V

Paul

in

delivers

a

talk

IN

N

PP

IN

N

Rome

Arg. 1
Flat features (Linear Kernel)

- To each example is associated a vector of 6 feature types

\[
\tilde{x} = (0, \ldots, 1, \ldots, 0, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 1, \ldots, 0, \ldots, 1, \ldots, 1)
\]

\[
\begin{align*}
\text{PT} & \quad \text{PTP} & \quad \text{HW} & \quad \text{PW} & \quad P & \quad V
\end{align*}
\]

- The dot product counts the number of features in common

\[
\tilde{x} \cdot \tilde{z}
\]
Automatic Predicate Argument Extraction

Deriving Positive/Negative example

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
Argument Classification Accuracy

![Graph showing the accuracy of argument classification for different training data percentages. The graph compares different methods: ST, SST, Linear, and PT.]}
# SRL in FrameNet: Results

<table>
<thead>
<tr>
<th>Eval Setting</th>
<th>Tree Kernels</th>
<th>Tree Kernels + PK</th>
<th>PK alone</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P$</td>
<td>$R$</td>
<td>$F_1$</td>
</tr>
<tr>
<td>BD</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BD Proj.</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BD+RC</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BD+RC Proj.</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>TK</td>
<td>TK</td>
<td>TK + PK</td>
</tr>
<tr>
<td>BD</td>
<td>.949</td>
<td>.652</td>
<td>.773</td>
</tr>
<tr>
<td>BD Proj.</td>
<td>.919</td>
<td>.631</td>
<td>.748</td>
</tr>
<tr>
<td>BD+RC</td>
<td>.697</td>
<td>.479</td>
<td>.568</td>
</tr>
<tr>
<td>BD+RC Proj.</td>
<td>.672</td>
<td>.462</td>
<td>.548</td>
</tr>
<tr>
<td></td>
<td>TKL</td>
<td>TKL</td>
<td>TKL + PK</td>
</tr>
<tr>
<td>BD</td>
<td>.938</td>
<td>.659</td>
<td>.774</td>
</tr>
<tr>
<td>BD Proj.</td>
<td>.906</td>
<td>.636</td>
<td>.747</td>
</tr>
<tr>
<td>BD+RC</td>
<td>.689</td>
<td>.484</td>
<td>.569</td>
</tr>
<tr>
<td>BD+RC Proj.</td>
<td>.663</td>
<td>.466</td>
<td>.547</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Eval Setting</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BD (nodes)</td>
<td>1.0</td>
<td>.732</td>
<td>.847</td>
</tr>
<tr>
<td>BD (words)</td>
<td>.963</td>
<td>.702</td>
<td>.813</td>
</tr>
<tr>
<td>BD+RC (nodes)</td>
<td>.784</td>
<td>.571</td>
<td>.661</td>
</tr>
<tr>
<td>BD+RC (words)</td>
<td>.747</td>
<td>.545</td>
<td>.630</td>
</tr>
</tbody>
</table>

Table 4.2: Results on the FrameNet dataset. Best configuration from Table 4.1, raised to 90% of training data for BD and RC.
**Current system: ACL 2010**

<table>
<thead>
<tr>
<th></th>
<th>Corpus</th>
<th>Predicates</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>training</strong></td>
<td>FN-BNC</td>
<td>134,697</td>
<td>271,560</td>
</tr>
<tr>
<td><strong>test</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>in-domain</em></td>
<td>FN-BNC</td>
<td>14,952</td>
<td>30,173</td>
</tr>
<tr>
<td><em>out-of-domain</em></td>
<td>NTI</td>
<td>8,208</td>
<td>14,422</td>
</tr>
<tr>
<td></td>
<td>ANC</td>
<td>760</td>
<td>1,389</td>
</tr>
</tbody>
</table>
In-domain SRL

- BNC tagging

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (StdDev)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Prior</td>
<td>43.9</td>
</tr>
<tr>
<td>Global Prior</td>
<td>67.7</td>
</tr>
<tr>
<td>Distributional</td>
<td>81.1 (±.52)</td>
</tr>
<tr>
<td>Backoff</td>
<td>84.6 (±.51)</td>
</tr>
<tr>
<td>Backoff+HMMRR</td>
<td>86.3 (±.17)</td>
</tr>
<tr>
<td>(Johansson&amp;Nugues, 2008)</td>
<td><strong>89.9</strong></td>
</tr>
</tbody>
</table>
Out-of-domain SRL
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?
Quest. Answering: an overall view

Figure 3: Retrieval Feedbacks in a Q/A System

(M. Pasca and S. Harabagiu, SIGIR 2001)
Question Classifier based on Tree Kernels

- 5500 training and 500 test questions [Li and Roth, 2004]
- Distributed on 6 categories:
  Abbreviations, Descriptions, Entity, Human, Location, and Numeric.
- Using the whole question parse trees
  - Two parsing paradigms: Constituent and Dependency
  - Example

  “What is an offer of direct stock purchase plan?”
Answer Type and Focus

• **Focus** is the word that expresses the relevant entity in the question
  - Used to select a set of relevant documents
  - ES: Where was **Mozart** born?

• **Answer Type** is the category of the entity to be searched as answer
  - PERSON, MEASURE, TIME PERIOD, DATE, ORGANIZATION, DEFINITION
  - ES: Where was **Mozart** born?
    - LOCATION
Answer Type and Focus

What famous communist leader died in Mexico City?

RULENAME: WHAT-WHO
OUTPUT: [“PERSON” ]

Answer type: PERSON
Focus: leader

This rule matches any question starting with what, whose first noun, if any, is a person (i.e. satisfies the person-p predicate)
Answer Type Taxonomies

Figure 1: Answer Type Taxonomy

(M. Pasca and S. Harabagiu, SIGIR 2001)
The dependency Tree

- “What is an offer of direct stock purchase plan”

PTs can be very effective, e.g.

[Plan [direct][purchase]] and [Plan [stock][purchase]]
## Question Classification results

<table>
<thead>
<tr>
<th>Parsers</th>
<th>Constituent</th>
<th>Dependency</th>
<th>BOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernels</td>
<td>SST</td>
<td>PT</td>
<td>SST</td>
</tr>
<tr>
<td></td>
<td>88.2</td>
<td>87.2</td>
<td>82.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90.4</td>
<td>Linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>86.3</td>
</tr>
</tbody>
</table>
Conclusions

- Tree kernels are a natural way to introduce syntactic information in natural language learning.
  - Very useful when few knowledge is available about the proposed problem.
  - e.g., manual feature design to encode predicate argument relations is complex
- Different forms of syntactic information require different tree kernels.
  - Collins and Duffy’s kernel (SST) useful for constituent parsing
  - The new Partial Tree kernel useful for dependency parsing
- Experiments on SRL and QC show that
  - PT and SST are efficient and very fast
  - Higher accuracy when the opportune kernel is used for the target task
Tree kernels for SVMs: References

- Accessible from the network:
Prestospace

• Scientific and technological objective:
  – to develop an integrated approach to audiovisual preservation and access, to produce sustainable assets with high cultural value and potential commercial use.

• Metadata Access and Delivery (MAD):
  – to generate, validate and deliver to the archive users metadata created by automatic and semi-automatic information extraction processes

• 32 partners, more than 8 EU member states

• Funding: about 3.3 M
MAD: an architectural Overview

MAD Documentation Factory
- Dynamic
- Quick
- Transient

Publication Platform
- Large Storage
- Public Access
- Search

Turnkey System

Export
Core Documentation Platform

GAMPS

Content Analysis
Speech To Text

Content Analysis
Shots-key frames

Content Analysis
Media Analysis

Semantic
Analysis

Manual
Annotation

Delivery

Web services

Core Platform

EMS
Essence and Management System

Rich Content

EDOB
MPEG7
PMETA
DC

MXF

JPG

Core Documentation Platform

WorkFlow Monitoring

EMS Monitoring
Content Analysis in MAD:

- Currently available GAMPs
  - Welcomer: demux MXF (RAI)
  - Content Analysis: Speech to text (RAI)
  - Content Analysis: Shots finder (RAI)
  - Content Analysis: Video segmenter (RAI)
  - Content Analysis: Media analiser (JRS)
  - Content Analysis: Stripe images (JRS)
  - Content Analysis: Camera motion (JRS)
  - Semantic Analysis: Text segmenter (Sheffield)
  - Semantic Analysis (Eng): Classifier, Named Entity extraction (Univ. of Sheffield)
  - Semantic Analysis (Ita): Classifier, Named Entity extraction (Univ. of Tor Vergata)
  - Semantic Analysis: Web Aligner (Univ. of Tor Vergata)
Semantic analysis (ItaSA GAMP)

- **Aim:**
  - to process textual transcriptions of multimedia content (e.g. radio and TV broadcast)
  - To provide information useful for retrieval, i.e. semantic metadata

- **Target Semantic Phenomena:**
  - Topical categories
  - Recognition/Classification of (internal) Named Entities (NE)
  - Alignment with external (Web) material
  - Recognition of events and NE participants
ItaSA GAMP: Architectural Overview
00:06:36: L'uragano di mallarme si è formato nel frattempo un'altre tempesta tropical.

00:06:41: Ha lasciato una riportazione delle sue distruzioni verso la Florida e l'uragano della corrente "willa" il dodicesimo ciclone di una stagione croce dell'atmosfera piu' di qualche lavatrice che si è la principale emergenza metereologica dell'anno. Le prime notizie della criacca e poco sopra 150 persone sono state costrette ad abbandonare le loro case a causa del terremoto.

00:06:52: Le prime notizie della criacca e poco sopra 150 persone sono state costrette ad abbandonare le loro case a causa del terremoto.

00:07:22: L'uragano di mallarme si è formato nel frattempo un'altre tempesta tropical.

00:07:44: Le prime notizie della criacca e poco sopra 150 persone sono state costrette ad abbandonare le loro case a causa del terremoto.

00:07:52: L'uragano di mallarme si è formato nel frattempo un'altre tempesta tropical.

00:07:59: L'uragano di mallarme si è formato nel frattempo un'altre tempesta tropical.

00:08:05: L'uragano di mallarme si è formato nel frattempo un'altre tempesta tropical.
Cross-language Retrieval (CLIR)

• Archive AV data span across different languages
• Retrieval should be concept rather than text oriented
• Source language (of queries) can be different from the target language (characterizing metadata)
CLIR in MAD

- **Technology**
  - Domain modelling via LSA (query expansion)
  - Word sense disambiguation *(Wordnet)*
  - Sense based translations
  - Ontological IDs
  - Merging different evidences (query translation)

- **Strength**
  - Fully automatic
  - Good accuracy
  - Portable across domains

- **Current limitations**
  - Resource Coverage *(Multiwordnet)*
 Blair calls on NATO member to contribute more troops to Afghanistan force.

NATO
- lemma -> nato
- possible translations:
  - [1.00000] [North_Atlantic_Treaty_Organisation, NATO] -> [n.a.t.o., organizzazione_del_trattato_nordatlantico]

member
- lemma -> member
- possible translations:
  - [member] -> [componente, membro]
  - [penis, phallos, member] -> [asta, fallo, membro, membro_virile, pene, verga]
  - [member] -> [appartenente, componente, iscritto, membro]
  - [extremity, appendage, member] -> [arto, estremita', membro]
  - [0.36195] [member] -> [membro]

troops
- lemma -> troop
- possible translations:
  - [1.00000] [military_personnel, soldiery, troops] -> [truppe]

Afghanistan
- lemma -> afghanistan
- possible translations:
  - [1.00000] [Afghanistan, Islamic_State_of_Afghanistan] -> [afghanistan]

force
- lemma -> force
- possible translations:
  - [force] -> [forza]
  - [0.150414] [military_unit, military_force, military_group, force] -> [arma]
  - [violence, force] -> [forza, violenza]
  - [force] -> [forza]
  - [effect, force] -> [effetto, forza]
  - [force, personnel] -> [personale]
  - [force, forcefulness, strength] -> [corpo, energia, forza, luna]

Blair
- lemma -> blair
-calls on
- lemma -> call_on
-contribute
- lemma -> contribute
CLIR GUI: Settings

Language specification
Query language: en ○ it ○ Target language: en ○ it

Insert the query:
Blair calls on NATO member to contribute more troops to Afghanistan force.

Query category

Common nouns and translations

Common nouns

+ NATO
  - lemma --> nato
CLIR GUI: translations

Query category

Common nouns and translations | Chaos NES | KIM NES | Sent XML | Received XML |
---|---|---|---|---|

Common nouns

- lemma -> nato
  + possible translations:
    * [1.00000] [North_Atlantic_Treaty_Organization, NATO] -> [n.a.t.o., organizzazione_del_trattato_nato]

- lemma -> member
  + possible translations:
    - [member] -> [componente, membro]
    - [penis, phallus, member] -> [asta, fallo, membro, membro_virile, pene, verga]
    - [member] -> [appartenente, componente, iscritto, membro]
    - [extremity, appendage, member] -> [arto, estremita', membro]
    * [0.361951] [member] -> [membro]

- lemma -> troop
  + possible translations:
    * [1.00000] [military_personnel, soldiery, troops] -> [truppa]

- lemma -> afghanistan
  + possible translations:
CLIR GUI: translation (2)

+ Afghanistan
  - lemma -> afghanistan
+ possible translations:
  * [1.000000] [Afghanistan, Islamic_State_of_Afghanistan] -> [afghanistan]
+ force
  - lemma -> force
+ possible translations:
  - [force] -> [forza]
  * [0.160414] [military_unit, military_force, military_group, force] -> [arma]
  - [violence, force] -> [forza, violenza]
  - [force] -> [forza]
  - [effect, force] -> [effetto, forza]
  - [force, personnel] -> [personale]
  - [force, forcefulness, strength] -> [corpo, energia, forza, lena]
+ Blair
  - lemma -> blair
+ calls_on
  - lemma -> call_on
+ contribute
  - lemma -> contribute
CLIR GUI: SA GAMP role

+ Blair
  - type -> Person
+ NATO
  - type -> Organization
+ Afghanistan
  - type -> Location


+ NATO
  - lemma --> nato
CLIR GUI: Results

Transcribed Query:

**Person:** Blair, **Organization:** Nato,

(n.a.t.o | "organizzazione del trattato nordatlantico"),

**Location:** Afghanistan, membro, truppe, arma
<table>
<thead>
<tr>
<th>Input Query</th>
<th>Berlusconi al parlamento sulla missione di guerra in Iraq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chaos NEs</td>
<td>Berlusconi [person] Iraq [paese]</td>
</tr>
<tr>
<td>KIM NEs</td>
<td>Iraq [Location]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nouns, Translations</th>
<th>Noun</th>
<th>Input language senses</th>
<th>Target language senses</th>
</tr>
</thead>
<tbody>
<tr>
<td>parlamento</td>
<td>parlamento</td>
<td>parliament</td>
<td></td>
</tr>
<tr>
<td>missione</td>
<td>delegazione, deputazione, missione, rappresentanza</td>
<td>deputation, commission, delegation, delegacy, mission</td>
<td></td>
</tr>
<tr>
<td></td>
<td>missione</td>
<td>mission, military mission</td>
<td></td>
</tr>
<tr>
<td>guerra</td>
<td>guerra</td>
<td>war, warfare</td>
<td></td>
</tr>
<tr>
<td></td>
<td>battaglia, combattimento, conflitto, guerra, lotta, scontro</td>
<td>battle, conflict, fight, engagement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>discordia, disunione, guerra, zizzania</td>
<td>discord, strife</td>
<td></td>
</tr>
<tr>
<td>guerra</td>
<td>guerra</td>
<td>strife</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output Query</th>
<th>Person: Berlusconi &amp; Location: Iraq &amp; parliament &amp; (deputation</th>
<th>commission</th>
<th>delegation</th>
<th>delegacy</th>
<th>mission) &amp; strife</th>
</tr>
</thead>
</table>

**CLIR GUI: Results (2)**
Prestospace: References

• Home Page: http://prestospace.org/