Distributed Tree Kernels and Distributed Convolution Kernels on Countable Sets

Fabio Massimo Zanzotto, Lorenzo Dell'Arciprete

ART Group

Dipartimento di Ingegneria dell'Impresa

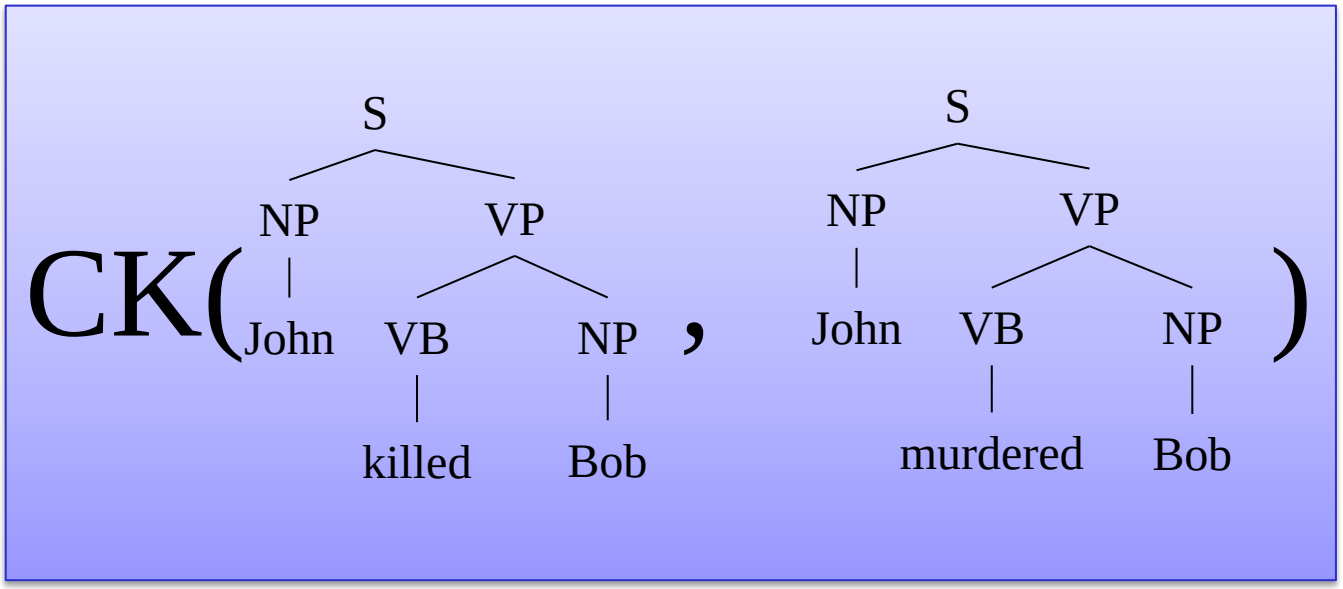
University of Rome "Tor Vergata"

Learning with Structures

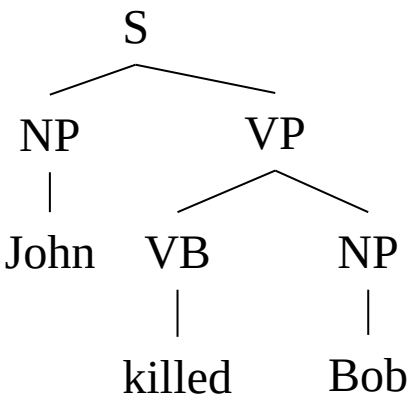
Classical Ingredients (to fully exploit structural information)

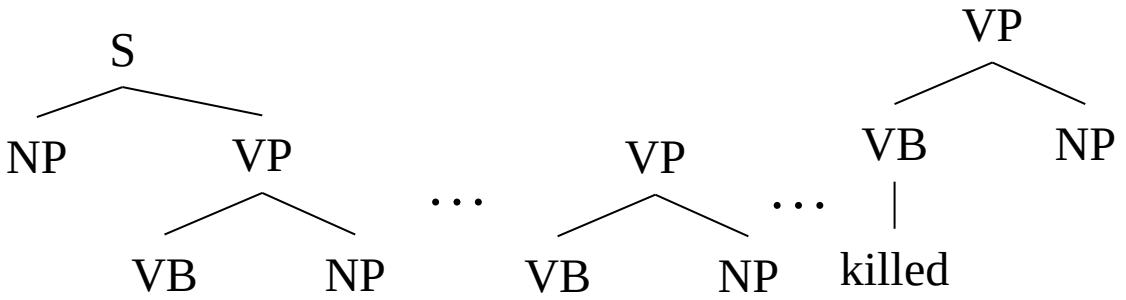
our favorite Kernel Machine

the desired Convolution Kernel $CK(x, y)$



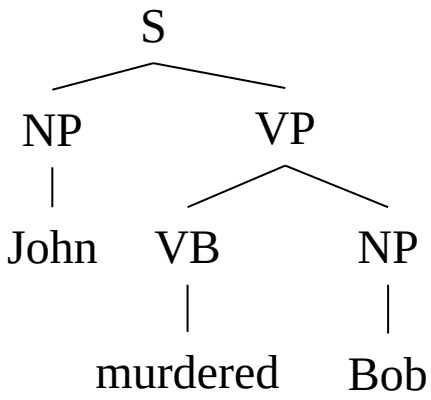
Convolution Kernels





(…,0,1,0,……,0,1,0,…,0,1,0,…….0,0,0,…)

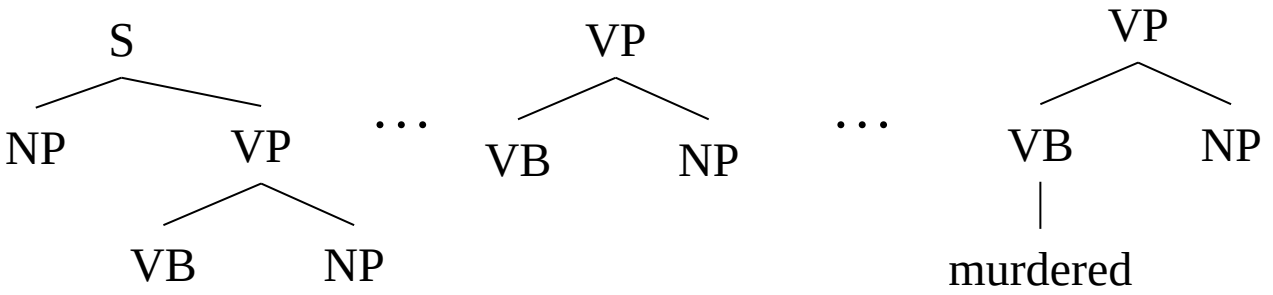
$CK(x, y)$



CKs perform weighted counts of common substructures,

hiding the dot product between the feature vectors

(…,0,1,0,……,0,1,0,…,0,0,0,…….0,1,0,…)



Learning with Structures

Classical Ingredients (to fully exploit structural information)

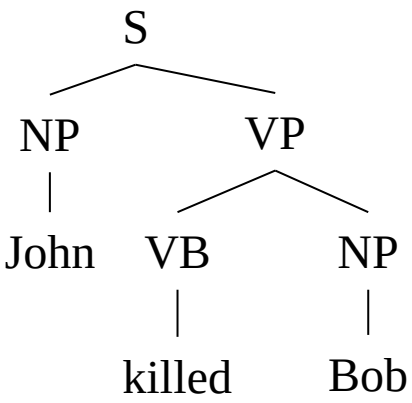
your favorite Kernel Machine

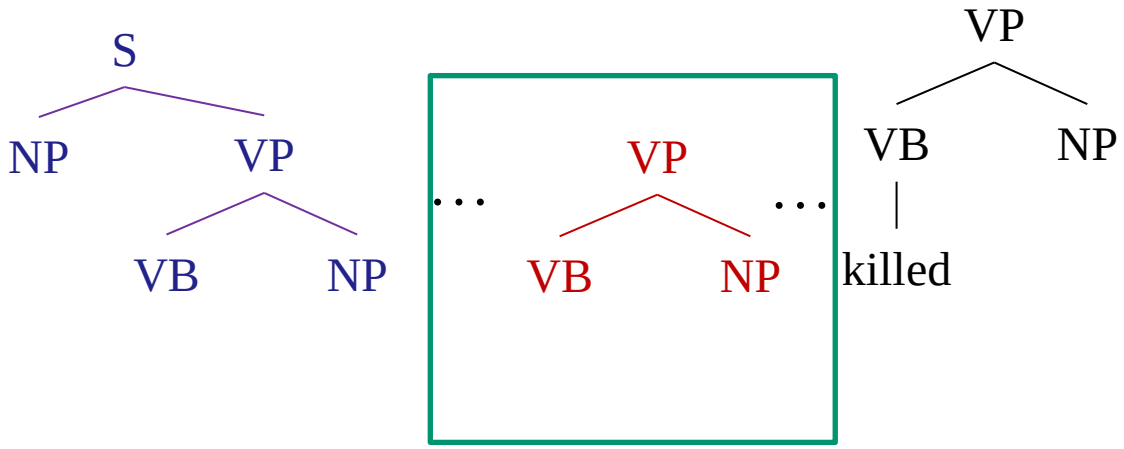
the desired Convolution Kernel $CK(x, y)$

Major limit

Treating structures with CK forces the learning method to be a kernel machine

Distributed Convolution Kernels

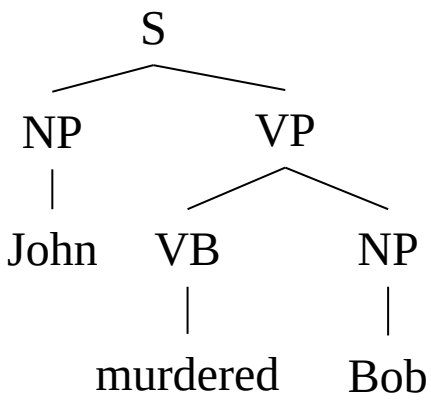




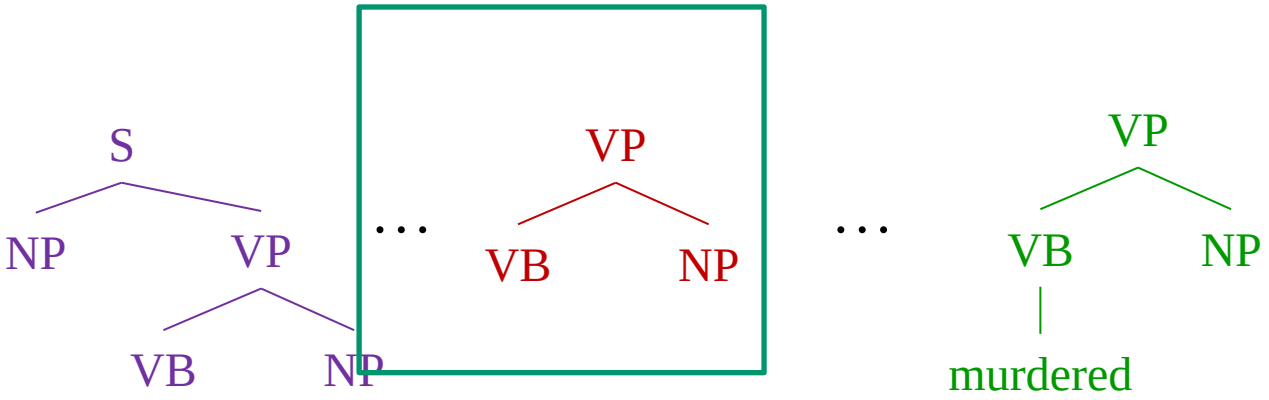
(0.239,-0.920,…,0.771)

$DCK(x, y)$

DCKs **directly** perform the dot product between the ***reduced*** feature vectors



(1.32,0.0280,…,-0.032)



Learning with Structures

Classical Ingredients (to fully exploit structural information)

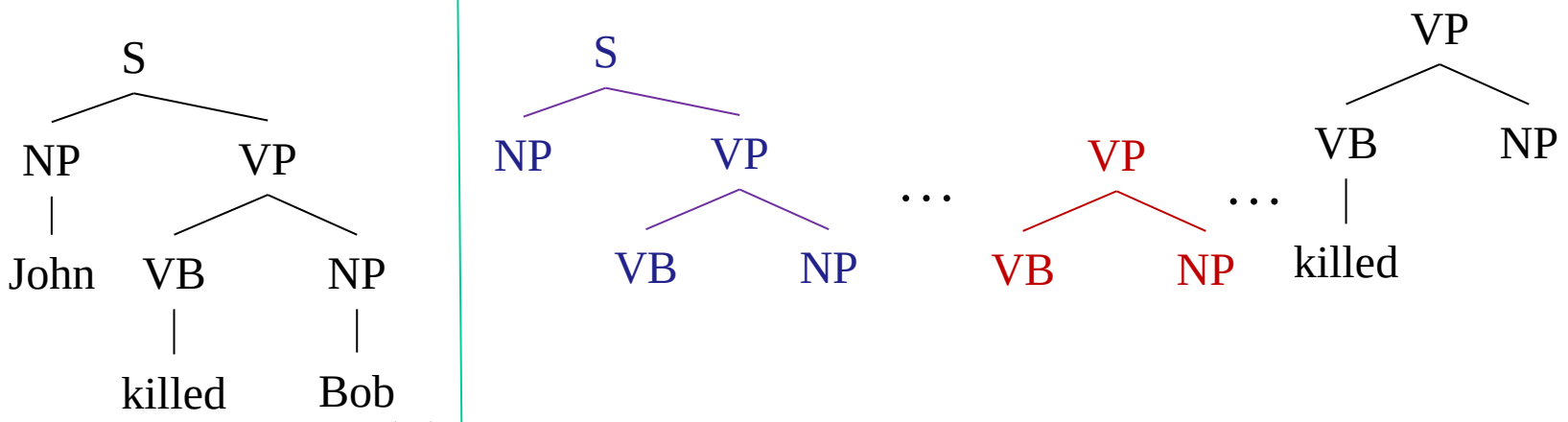
- your favorite Kernel Machine
- the desired Convolution Kernel $CK(x, y)$

New Ingredients

- the ***Distributed Convolution Kernel*** approximating the desired $CK(x, y)$
- a ***linear version*** of your favorite Kernel Machine

Linear versions of Kernel Machines are extremely fast (e.g., Pegasos)!

Distributed Convolution Kernels



$z \qquad S(z) = \{c_i \quad \dots \quad c_j \quad \dots \quad c_k\}$

$D : Structures \rightarrow \mathbb{R}^d \qquad \Gamma : Structures \rightarrow \mathbb{R}^d$

Γ is a **Compositional Function**

$$D(z) = \sum_{c \in S(z)} \omega_c \Gamma(c)$$

$$c_j = \Gamma(c_j) \qquad c_i = \Gamma(c_i) \qquad c_k = \Gamma(c_k)$$

$$\begin{bmatrix} 0.00032132 \\ -0.00039842 \\ \vdots \\ -0.00921011 \end{bmatrix} \quad \begin{bmatrix} -0.00136979 \\ 0.00043675 \\ \vdots \\ -0.00084673 \end{bmatrix} \quad \dots \quad \begin{bmatrix} 0.00154736 \\ -0.00075940 \\ \vdots \\ -0.00056302 \end{bmatrix} \quad \dots \quad \begin{bmatrix} 0.00301736 \\ -0.00098940 \\ \vdots \\ 0.00021220 \end{bmatrix}$$

Names for the «Distributed» World

As we are encoding structures in small vectors, we follow the tradition of ***distributed structures*** *(Plate, 1994)*

Distributed **C**onvolution **K**ernels (DCK)

$$DCK(x,y) = \langle D(x), D(y) \rangle$$

Distributed Convolution Structures

$$D(z) = \sum_{c \in S(z)} \omega_c \Gamma(c)$$

$$\begin{bmatrix} 0.00032132 \\ -0.00039842 \\ \vdots \\ -0.00921011 \end{bmatrix}$$

Distributed Structures

$$c_j = \Gamma(c_j) \qquad c_i = \Gamma(c_i) \qquad c_k = \Gamma(c_k)$$

$$\begin{bmatrix} -0.00136979 \\ 0.00043675 \\ \vdots \\ -0.00084673 \end{bmatrix} \quad \begin{bmatrix} 0.00154736 \\ -0.00075940 \\ \vdots \\ -0.00056302 \end{bmatrix} \quad \begin{bmatrix} 0.00301736 \\ -0.00098940 \\ \vdots \\ 0.00021220 \end{bmatrix}$$

In the rest of the presentation

- **DCK:** expected properties and challenges
- **Distributed Structures:** the function Γ
- **Distributed Convolution Structures**
- **Some facts:** 2 Lemmas and 1 Theorem to show expected properties
- **Implemented DCKs**
- A taste of **experimental investigation**
- Conclusions and Future Work

DCK: Expected properties and challenges

Given that:

$$DCK(x,y) = \langle D(x), D(y) \rangle = \sum_{c \in S(x), b \in S(y)} \omega_c \omega_b \langle \Gamma(c), \Gamma(b) \rangle$$

We expect

- **Distributed Structures** have the following property:

$$\delta(c_1, c_2) - \varepsilon < \langle \Gamma(c_1), \Gamma(c_2) \rangle < +\delta(c_1, c_2) + \varepsilon$$

- Each **Convolution Kernel** (CK) has an associated **Distributed Convolution Structure** (D) such that:

$$CK(a,b) - \varepsilon J < \langle D(a), D(b) \rangle < CK(a,b) + \varepsilon J$$

Definition: Structured Object

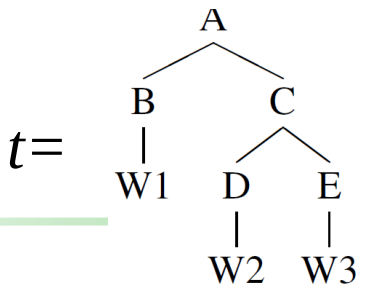
A ***structured object*** $x \in X$ is either:

- a terminal object (in the set of ground objects G_X)
- an object that can be decomposed in M parts

$$x = (x_1, x_2, \ldots, x_M) \in X$$

where x_i are again structured objects

e.g. a tree t is $t = (r, c_1, c_2, \ldots, c_k)$ r is the root and c_i is the i-th subtree:



$\bar{t}$ =(A,(B W1),(C (D W2)(E W3)))

Distributed Structures: the compositional function Γ

Basic elements

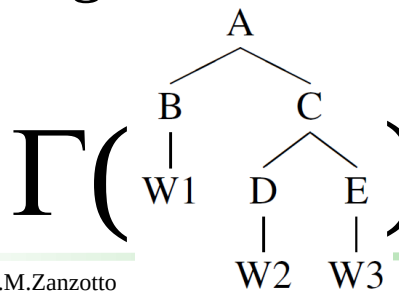
g_X set of nearly orthogonal random vectors

$\odot$ a basic vector composition function

A ***distributed structure*** is recursively defined as:

$$\Gamma(x) = \begin{cases} \gamma(x) = x \in g_X & \text{if } x \text{ is a terminal object} \\ \bigodot_i \Gamma(x_i) & \text{otherwise} \end{cases}$$

e.g.



$$\Gamma(\quad) = (\vec{A} \odot (\vec{B} \odot \overrightarrow{W1}) \odot (\vec{C} \odot (\vec{D} \odot \overrightarrow{W2}) \odot (\vec{E} \odot \overrightarrow{W3})))$$

Distributed Structures: the compositional function Γ

Properties of the **Ideal function** $\odot$

1. Non-commutativity with a very high degree k
2. Non-associativity
3. Bilinearity

Approximation

4. $\|a \odot b\| = \|a\|\|b\|$
5. $|\langle t, a \rangle| < \varepsilon$ if $t \neq a$
6. $|\langle a \odot b, c \odot d \rangle| < \varepsilon$ if $|\langle a, c \rangle| < \varepsilon$ or $|\langle b, d \rangle| < \varepsilon$

With the above properties, we proved that

$$\delta(c_1, c_2) - \varepsilon < \langle \Gamma(c_1), \Gamma(c_2) \rangle < +\delta(c_1, c_2) + \varepsilon$$

Distributed Convolution Structures

Definition

A function $D : X \to \mathbb{R}^d$ is a Distributed Convolution Structure if:

$$D(x) = \begin{cases} \omega_x \gamma(x) & \text{if } x \text{ is a terminal object} \\ \sum_{\overline{\mathbf{x}} \in R(x)} \bigodot_{i=1}^{M} D_i(x_i) & \text{otherwise} \end{cases}$$

where $D_i(\cdot)$ a distributed convolution structure itself

The final theorem

Theorem A convolution kernel over countable sets:

$$CK(x,y)$$

has an associated distributed convolution structures:

$$D(z)$$

such that:

$$CK(a,b) - \varepsilon \sum_{\substack{a \in S(z) \\ b \in S(y)}} \omega_a \omega_b < \langle D(a), D(b) \rangle < CK(a,b) + \varepsilon \sum_{\substack{a \in S(z) \\ b \in S(y)}} \omega_a \omega_b$$

Implemented DCKs

- Distributed Tree Kernels (DTK) (Zanzotto&Dell'Arciprete, ICML 2012)
- Distributed Subpath Kernels
- Distributed Route Kernels
- Distributed String Kernels
- Distributed Partial Tree Kernels

- ***New!*** **Distributed Smoothed Tree Kernel**

A taste of the experimental investigation

- Choosing a real function to approximate the ideal function $\odot$
- Comparing Convolution Kernels and Distributed Convolution Kernels
- We show here:
  - **Distributed Tree Kernels**
  - **Distributed Smoothed Tree Kernels**

Towards the reality: Approximating $\odot$

- $\odot$ is an **ideal** function!
- Proposed approximations:
  - Shuffled normalized element-wise product $\boxtimes$

$$\vec{a} \boxtimes \vec{b} = \gamma \cdot p_1(\vec{a}) \otimes p_2(\vec{b})$$

  - Shuffled circular convolution $\boxdot$

$$\vec{a} \boxdot \vec{b} = p_1(\vec{a}) \odot p_2(\vec{b})$$

It is possible to show that properties of $\odot$ statistically hold for the two approximations

Direct Analysis: DTK

- Spearman's correlation between DTK and TK values
- Test trees taken from QC corpus and RTE corpus

	QC		RTE	
λ	$DTK_{\boxtimes}$	$DTK_{\boxdot}$	$DTK_{\boxtimes}$	$DTK_{\boxdot}$
0.2	0.993	0.994	0.997	0.998
0.4	0.980	0.989	0.990	0.961
0.6	0.908	0.880	0.890	0.350
0.8	0.644	0.377	0.469	0.039
1.0	0.316	0.107	0.169	0.000

Beyond "Countable Set"

- With this method:
  - Discrete similarity between structures
  - Either two subtrees are the same or they aren't

- Beyond this restriction: **General Convolution Kernel**:
  - More complex similarity measure
  - Distributed Smoothed Tree Kernel (Ferrone&Zanzotto, Coling, 2014)

$$D(x) = \sum_{c \in S(x)} \omega_c \Gamma(c) \otimes \mathbf{w_n}$$

Add semantic part

Direct Analysis: DSTK

- Spearman's correlation between DSTK and STK values
- Test trees taken from RTE and STS corpus

Dataset	
RTE1	0.87
RTE2	0.84
RTE3	0.91
RTE5	0.84
Average	**0.87**

Dataset	
Headl	0.90
FNWN	0.65
OnWN	0.96
SMT	0.77
Average	**0.82**

Conclusions: a new way of Learning with Structures

Classical Ingredients (to fully exploit structural information)

- your favorite Kernel Machine
- the desired Convolution Kernel $CK(x, y)$

New Ingredients

- the ***Distributed Convolution Kernel*** approximating the desired $CK(x, y)$
- a ***linear version*** of your favorite Kernel Machine

Linear versions of Kernel Machines are extremely fast (e.g., Pegasos)!

Future Work

- Formally proving the observed properties of the shuffled circular convolution

- Expand the theory of General Distributed Convolution Kernels

References

References

Zanzotto&Dell'Arciprete, *Distributed Tree Kernels*, Proceedings of ICML, 2012

Zanzotto&Dell'Arciprete, *Distributed Convolution Kernels on Countable Sets*, JMLR (accepted upon minor revisions)

Zanzotto&Ferrone, *Towards Syntax-aware Compositional Distributional Semantic Models* , Proceedings of Coling, 2014