Bonaventura Coppola, **Alessandro Moschitti** and **Daniele Pighin**
Department of Computer Science and Engineering
University of Trento, Italy
{coppola,moschitti,pighin}@disi.unitn.it

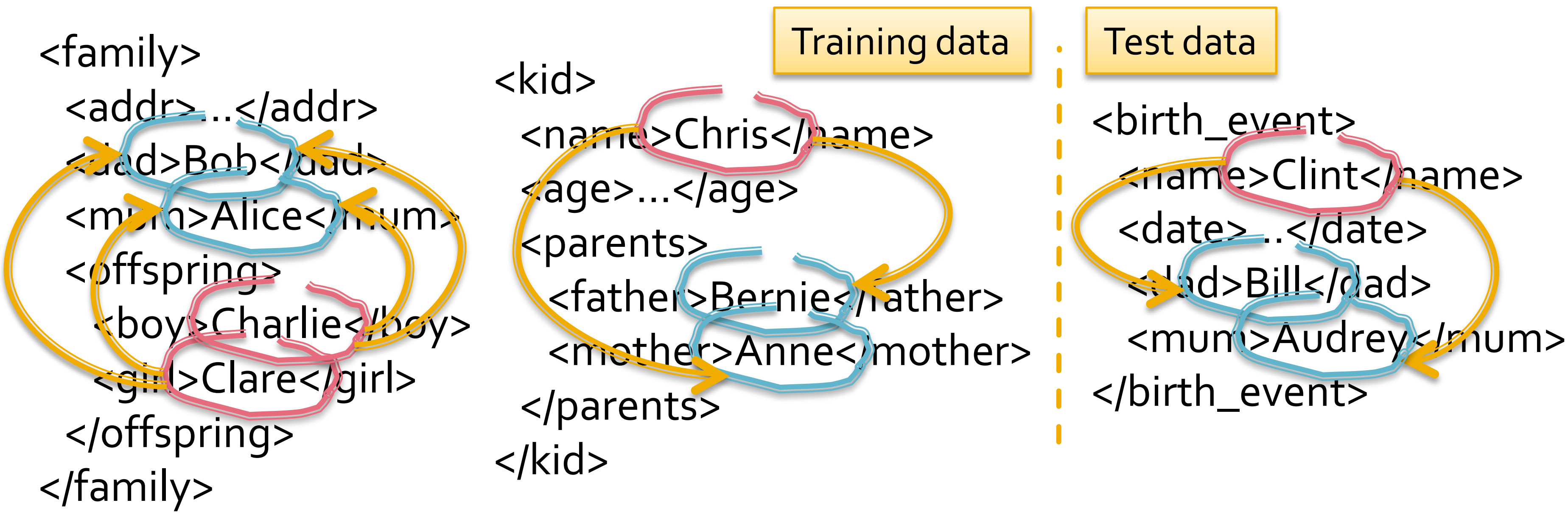
Generalized Framework for Syntax-based Relation Mining

Summary

- A framework for mining functional relations in structured data
- Supervised learning based on Support Vector Machines and Kernel Methods
- Structured features in the learning algorithm:
  - shorter development time
  - increased portability

Relations in Structured Data

- Supervised learning context



- Different syntactic representations ⇔ Same semantics
  Example: **is_child_of**(**Child**, **Parent**)
- **Child** and **Parent** are *roles* of the **is_child_of** *relation*

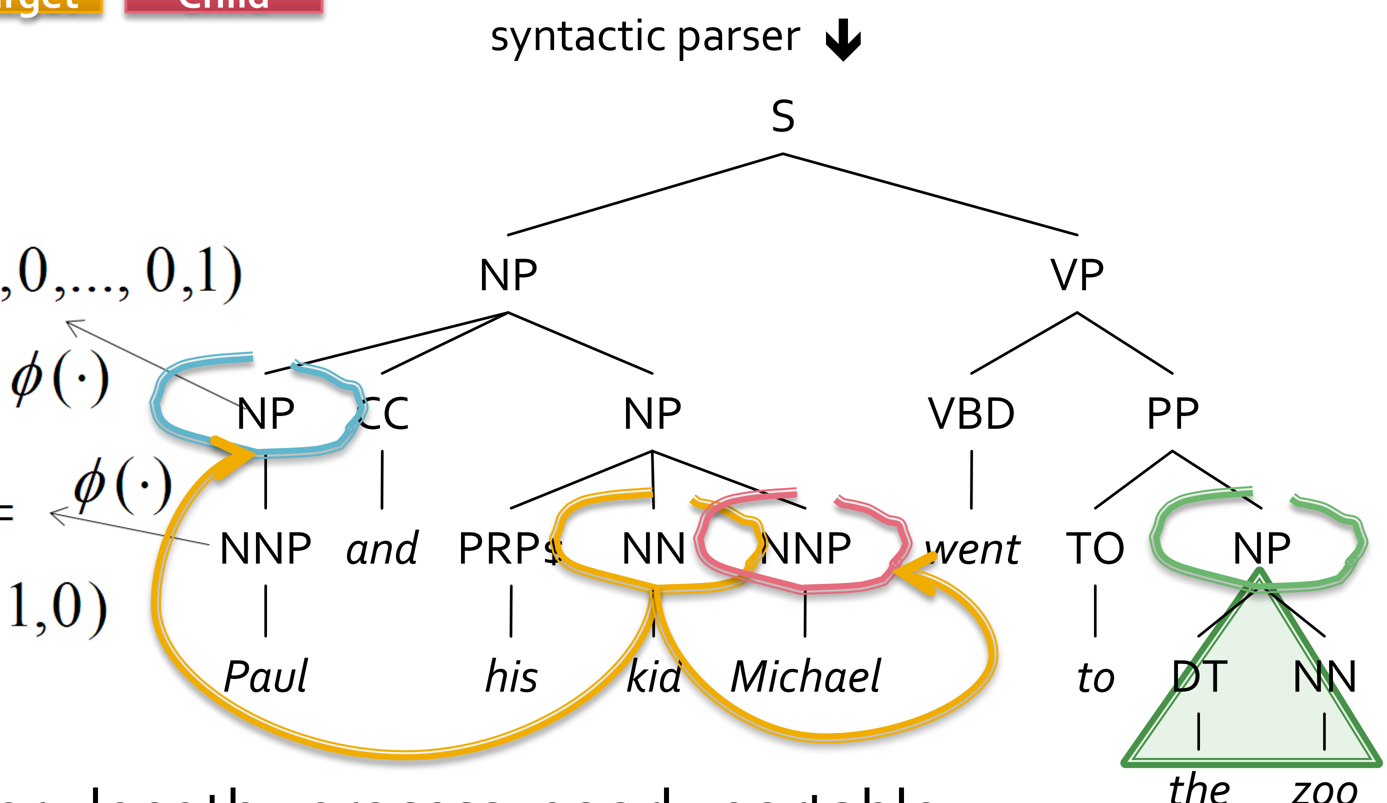
Relations in Natural Language

Paul and his kid Michael went to the zoo

Parent Target Child

$$\phi(\text{NP}) = \vec{x}_1 =$$
$$= (0,1,0,0,1,1,0,0,\ldots,0,1)$$

$$\phi(\text{NNP}) = \vec{x}_2 =$$
$$= (0,1,1,0,0,\ldots,1,0)$$



Problems: very lengthy process, poorly portable

Support Vector Machines and Kernel Functions

- Kernel trick: evaluate pair-wise similarity in implicit spaces

$$f(\vec{x}) = \operatorname{sgn}\left(\sum_{i=1..\ell} y_i \alpha_i \vec{x}_i \cdot \vec{x} + b\right) = \operatorname{sgn}\left(\sum_{i=1..\ell} y_i \alpha_i \,\mathrm{k}(o_i, o) + b\right)$$

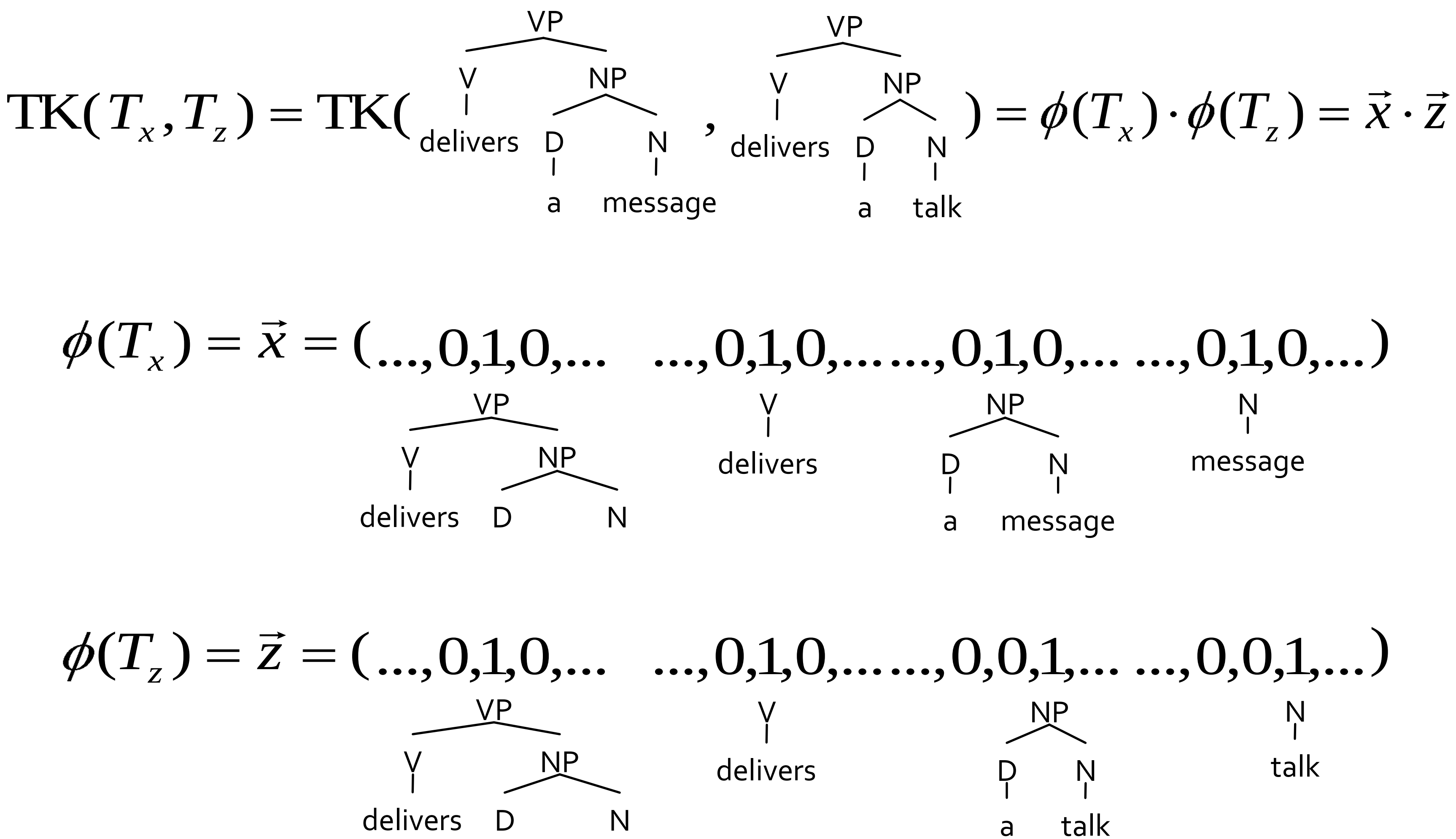
- **Polynomial** kernel: conjunction of independent linear features
- **Tree** Kernel:

$$\mathrm{TK}(T_x, T_z) = \phi(T_x) \cdot \phi(T_z) = \vec{x} \cdot \vec{z}$$

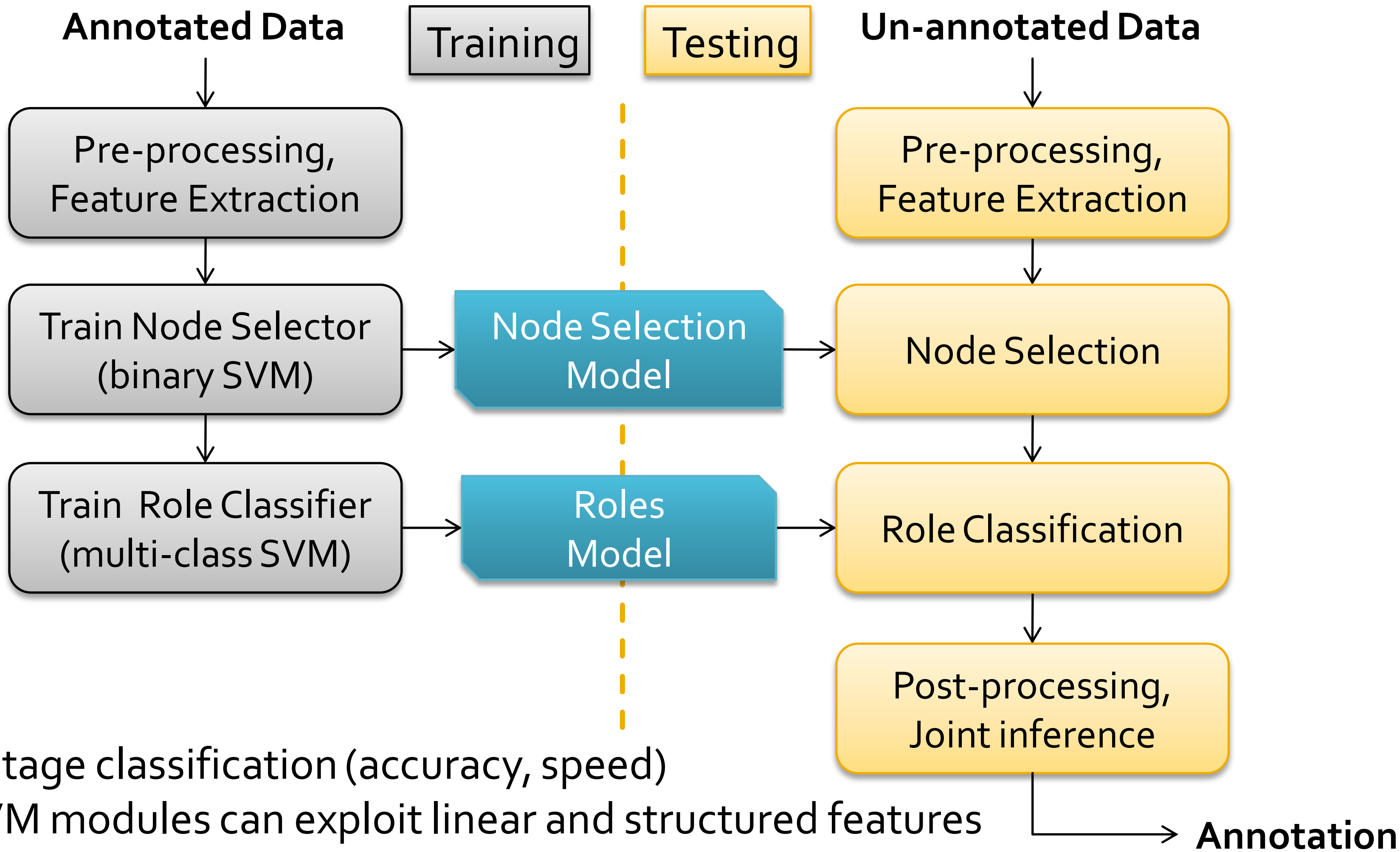
$$f(T) = \operatorname{sgn}\left(\sum_{i=1..\ell} y_i \alpha_i \,\mathrm{TK}(T_i, T) + b\right)$$

  - *T*: **structured feature** describing candidates
  - Measures the similarity between two trees by counting the number of common substructures
  - Tree Kernels can trigger automatic feature selection

Tree Kernel Evaluation



Relational Mining Architecture (RMA)



- 2 stage classification (accuracy, speed)
- SVM modules can exploit linear and structured features

Single-domain Relation Extraction

- PropBank:
  - Lexical-semantic resource used in Semantic Role Labeling
  - Semantic annotation layer over the Penn TreeBank
  - 59 distinct and general role labels
  - 43,616 annotated sentences
- Example:
  - **[Paul] A0** *clouded* [**his senses**] **A1** [**with heavy doses of children's cough syrup**] **A2**
  - Labels can be mapped to general semantic roles (Agent, Instrument, …)

Frame Semantics and Multi-domain Scenario

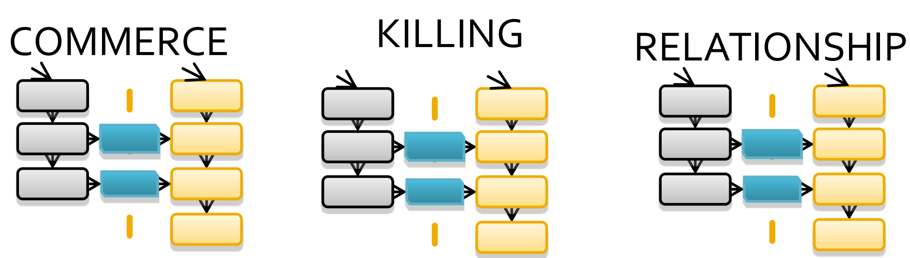
- Frame: *abstract representation of an event, situation, or property*
- Example: COMMERCE_SCENARIO
  Core Elements: BUYER, GOODS, MONEY, SELLER
  non-Core Elements: MANNER, MEANS, PURPOSE, RATE
  Lexical Units: *buy, sell, purchase, acquisition, …*
  Subframes: COMMERCIAL_TRANSACTION
- Instance: "*Ralemberg said [he]SELLER already had a [buyer]BUYER [for the wine]GOODS*"
- The FrameNet Project developed definitions for ~800 Frames and ~4000 Frame-dependent Elements over 135,000 sentences.

Multi-Frame Mining Architecture

- Multi-Frame Architecture built over multiple basic RMA modules
- Advantages:
  - Deeper Semantic Representation with *local* Frame-based definitions
  - Easier porting to new application domains (new Frames)
- Allows for Selective Sharing of Learning Models

Classification Steps:

1. Target Word Selection (TS)
2. Frame Disambiguation (FD)
3. Node Selection (NS)
4. Role Classification (RC)



- Typical case: shared NS, and dedicated RC models

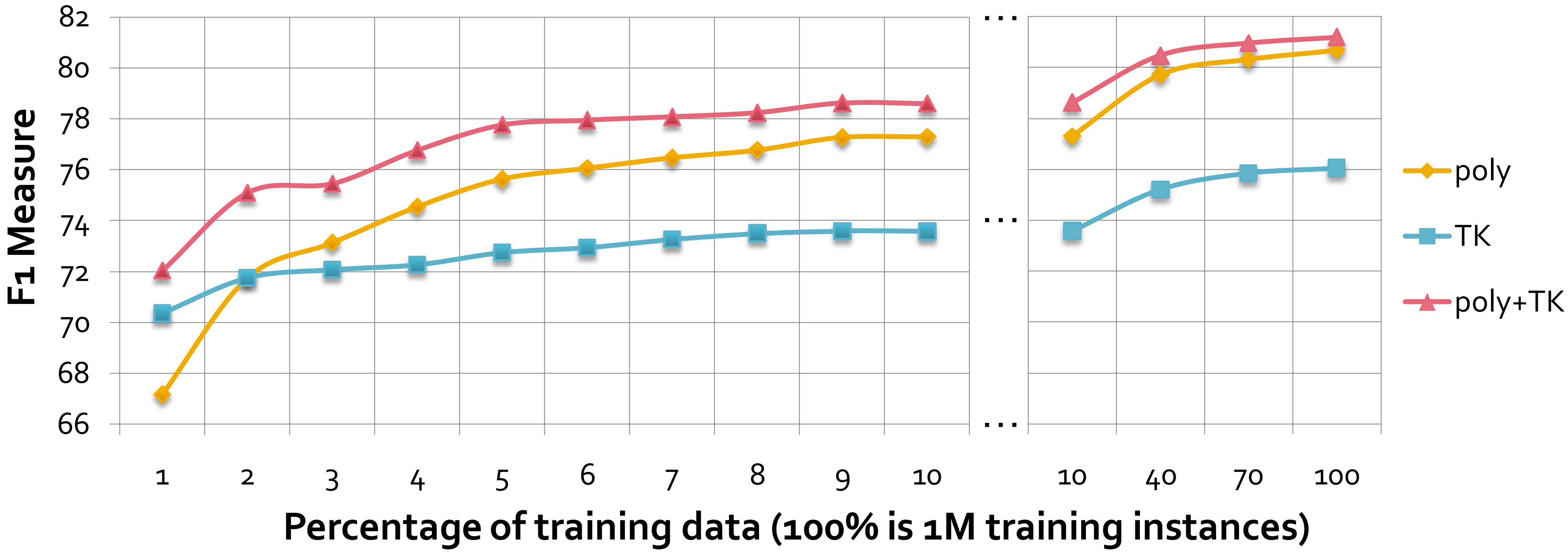
Evaluation: PropBank NS and RC

- Training: 1M instances for NS
- Configuration: poly+TK
- Test set: 270K instances
- Standard benchmark (CoNLL 2005)

Task	P	R	F1
NS	82.23%	80.83%	81.52
NS+RC	76.55%	75.24%	75.89
Joint inference	80.16%	74.54%	77.25

- The joint model can compensate for local classifier errors
- Uses TK based re-ranking of predicate argument structures

Evaluation: PropBank NS



- Evaluation on development set (149K nodes)
- TK+poly always improves over poly
- With very small training set, TK better than poly

Evaluation: FrameNet

- Evaluation data set: FrameNet 1.3
  - 135,293 parsed sentences
  - Training NS: 2%, Training RC: 90%, Test: 1%

	poly			TK			poly+TK		
Eval setting	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
NS (nodes)	0.89	0.68	0.77	0.95	0.65	0.77	0.92	0.7	**0.79**
NS (words)	0.85	0.65	0.74	0.92	0.63	0.75	0.88	0.67	**0.76**
NS+RC (nodes)	0.65	0.5	0.57	0.7	0.48	0.57	0.68	0.52	**0.59**
NS+RC (words)	0.63	0.48	0.54	0.67	0.46	0.55	0.65	0.5	**0.56**

- poly achieves better recall, TK better precision
- The poly+TK combination performs best

Evaluation: FrameNet (enhanced)

- DataSet: FrameNet 1.3
  - 135,293 parsed sentences
  - Training set increased to 90% for both NS and RC

	poly+TK (enhanced)		
Eval Setting	**P**	**R**	**F1**
NS (nodes)	1	0.73	**0.85**
NS (words)	0.96	0.7	**0.81**
NS+RC (nodes)	0.78	0.57	**0.66**
NS+RC (words)	0.75	0.55	**0.63**

- Results reach the state-of -the-art

Conclusions

- We presented a general framework for mining relations from tree-structured information in different domains
- Flexibility is achieved by Tree Kernels and structured machine learning features
- The framework has been tested on semantic analysis of parsed natural language texts, with state-of-the-art results
- Further experiments on different domains are planned, like protein active site detection

Bonaventura Coppola, **Alessandro Moschitti** and **Daniele Pighin**
Department of Computer Science and Engineering
University of Trento, Italy
{coppola,moschitti,pighin}@disi.unitn.it

Generalized Framework for Syntax-based Relation Mining

Thanks!

For more details, please meet us at our poster site!