Natural Language Processing and Information Retrieval

Part II: Structured Output

Alessandro Moschitti

Department of information and communication technology
University of Trento
Email: moschitti@dit.unitn.it

Output Label Sets

Simple Structured Output

- We have seen methods for: binary Classifier or multiclassifier single label
- Multiclass-Multilabel is a structured output, i.e. a label subset is output

From Binary to Multiclass classifiers

- Three different approaches:
- **ONE-vs-ALL (OVA)**
  - Given the example sets, {E1, E2, E3, …} for the categories: {C1, C2, C3,…} the binary classifiers: {b1, b2, b3,…} are built.
  - For b1, E1 is the set of positives and E2∪E3 ∪… is the set of negatives, and so on
  - <u>For testing</u>: given a classification instance x, the category is the one associated with the maximum margin among all binary classifiers

From Binary to Multiclass classifiers

- **ALL-vs-ALL (AVA)**
  - Given the examples: {E1, E2, E3, …} for the categories {C1, C2, C3,…}
    - build the binary classifiers:

      {b1_2, b1_3,…, b1_n, b2_3, b2_4,…, b2_n,…,bn-1_n}
    - by learning on E1 (positives) and E2 (negatives), on E1 (positives) and E3 (negatives) and so on…
  - For testing: given an example x,
    - all the votes of all classifiers are collected
    - where b_{E1E2} = 1 means a vote for C1 and b_{E1E2} = -1 is a vote for C2
  - Select the category that gets more votes

From Binary to Multiclass classifiers

- **Error Correcting Output Codes** (ECOC)
  - The training set is partitioned according to binary sequences (codes) associated with category sets.
    - For example, 10101 indicates that the set of examples of C1,C3 and C5 are used to train the C_{10101} classifier.
    - The data of the other categories, i.e. C2 and C4 will be negative examples
  - <u>In testing</u>: the code-classifiers are used to decode one the original class, e.g.

    $C_{10101} = 1$ and $C_{11010} = 1$ indicates that the instance belongs to C1

That is, the only one consistent with the codes

Designing Global Classifiers

- Each class has a parameter vector (w_k, b_k)
- x is assigned to class k iff

$$w_k^\top x + b_k \geq \max_j w_j^\top x + b_j$$

- For simplicity set b_k=0
  (add a dimension and include it in w_k)
- The goal (given separable data) is to choose w_k s.t.

$$\forall (x^i, y^i), \quad {w_{y^i}}^\top x^i \geq \max_j w_j^\top x^i$$

Multi-class SVM

Primal problem: QP

$$\min_{w_1,\dots,w_K} \quad \frac{1}{2}\|(w_1,\dots,w_K)\|^2 + C\sum_{ik}\xi_{ik}$$

$$\text{s.t.} \quad \forall (i,k), \quad {w_{y^i}}^\top x^i - w_k^\top x^i \geq \mathbf{1}\{k \neq y^i\} - \xi_{ik}$$

Structured Output Model

- Main idea: define scoring function which **decomposes** as sum of features scores k on "**parts**" p:

$$score(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \mathbf{w}^\top \Phi(\mathbf{x}, \mathbf{y}) = \sum_{k,p} w_k^\top \phi_k(\mathbf{x}_p, \mathbf{y}_p)$$

- Label examples by looking for max score:

$$prediction(\mathbf{x}, \mathbf{w}) = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} score(\mathbf{x}, \mathbf{y}, \mathbf{w})$$

space of feasible outputs

- Parts = nodes, edges, etc.

Structured Perceptron

Inputs: Training set (x_i, y_i) for $i = 1 \ldots n$

Initialization: $\mathbf{W} = 0$

Define: $F(x) = \operatorname{argmax}_{y \in \mathbf{GEN}(x)} \mathbf{\Phi}(x, y) \cdot \mathbf{W}$

Algorithm: For $t = 1 \ldots T$, $i = 1 \ldots n$

$z_i = F(x_i)$

If $(z_i \neq y_i)$ $\quad \mathbf{W} = \mathbf{W} + \mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i)$

Output: Parameters $\mathbf{W}$

(Averaged) Perceptron

For each datapoint $\mathbf{x}^i$

Predict: $$\hat{\mathbf{y}}_i = \arg\max_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}_t^\top \Phi(\mathbf{x}^i, \mathbf{y})$$

Update: $$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \underbrace{\Big(\Phi(\mathbf{x}, \mathbf{y}^i) - \Phi(\mathbf{x}^i, \hat{\mathbf{y}}_i)\Big)}_{\text{update if } \hat{\mathbf{y}}_i \neq \mathbf{y}^i}$$

Averaged perceptron: $$\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}_t$$

Example: multiclass setting

Predict: $\hat{y}_i = \arg\max_y \; w_y^\top x^i$

Update: if $\hat{y}_i \neq y^i$ then

$$w_{y^i,t+1} = w_{y^i,t} + \alpha x^i$$
$$w_{\hat{y}_i,t+1} = w_{\hat{y}_i,t} - \alpha x^i$$

Feature encoding:

$$\begin{aligned}
\Phi(\mathbf{x}^i, y=1)^\top &= [\mathbf{x}^{i\top} \; 0 \ldots 0] \\
\Phi(\mathbf{x}^i, y=2)^\top &= [0 \; \mathbf{x}^{i\top} \ldots 0] \\
&\vdots \\
\Phi(\mathbf{x}^i, y=K)^\top &= [0 \; 0 \ldots \mathbf{x}^{i\top}] \\
\mathbf{w}^\top &= [w_1^\top \; w_2^\top \ldots w_K^\top]
\end{aligned}$$

Predict: $\hat{\mathbf{y}}_i = \arg\max_{\mathbf{y} \in \mathcal{Y}} \; \mathbf{w}_t^\top \Phi(\mathbf{x}^i, \mathbf{y})$

Update: $\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \underbrace{\left(\Phi(\mathbf{x}, \mathbf{y}^i) - \Phi(\mathbf{x}^i, \hat{\mathbf{y}}_i)\right)}_{\text{update if } \hat{\mathbf{y}}_i \neq \mathbf{y}^i}$

Output of Ranked Example List

Support Vector Ranking

$$\begin{cases} min \quad \frac{1}{2}||\vec{w}|| + C\sum_{i=1}^{m}\xi_i^2 \\ y_k(\vec{w}\cdot(\vec{x_i}-\vec{x_j})+b) \geq 1-\xi_k, \quad \forall i,j=1,..,m \\ \xi_k \geq 0, \quad k=1,..,m^2 \end{cases}$$

y_k = 1 if $rank(\vec{x_i}) > rank(\vec{x_j})$, 0 otherwise, where $k = i \times m + j$

- Given two examples we build one example (x_i , x_j)

Concept Segmentation and Classification task

- Given a transcription, i.e. a sequence of words, chunk and label subsequences with concepts
- Air Travel Information System (ATIS)
  - Dialog systems answering user questions
  - Conceptually annotated dataset
  - Frames

An example of concept annotation in ATIS

- User request: ***list TWA flights from Boston to Philadelphia***

$$\underbrace{list}_{null}\ \underbrace{TWA}_{airline\_code}\ \underbrace{flights}_{null}\ \underbrace{from}_{null}\ \underbrace{Boston}_{fromloc.city}\ \underbrace{to}_{null}\ \underbrace{Philadelphia}_{toloc.city}$$

- The concepts are used to build rules for the dialog manager (e.g. actions for using the DB)
  - from location
  - to location
  - airline code

list flights from boston to Philadelphia	
FRAME:	FLIGHT
	FROMLOC.CITY = boston
	TOLOC.CITY = Philadelphia

Our Approach
(Dinarelli, Moschitti, Riccardi, SLT 2008)

- Use of Finite State Transducer to generate word sequences and concepts
- Probability of each annotation

$\Rightarrow$ *m* best hypothesis can be generated

- Idea: use a discriminative model to choose the best one
  - Re-ranking and selecting the top one

Experiments

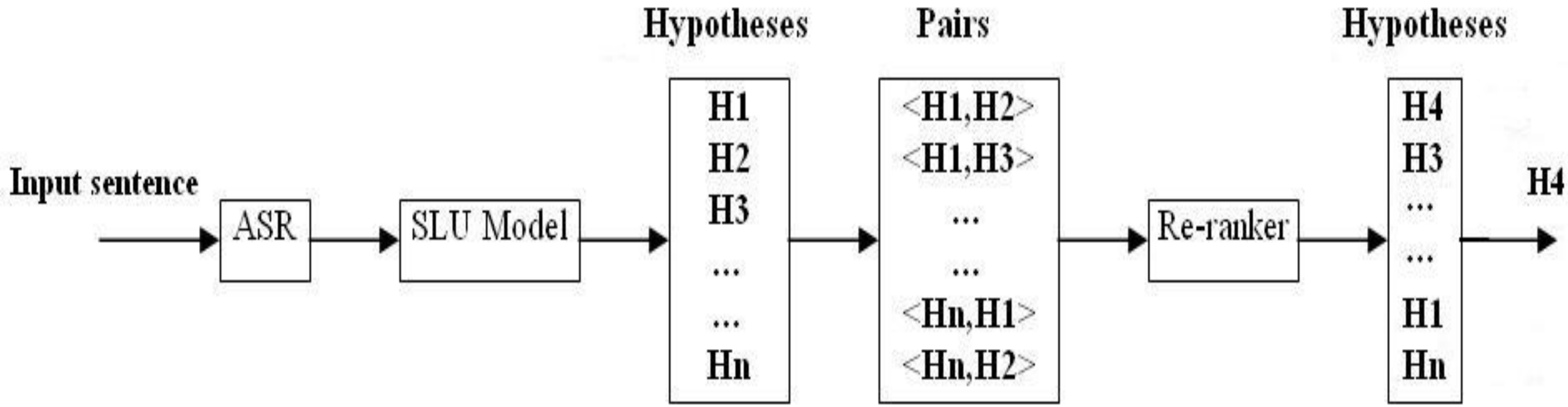
- Luna projects' Corpus Wizard of OZ

Corpus LUNA	Training set		Test set	
	words	concepts	words	concepts
Dialogs	183		67	
Turns	1,019		373	
Tokens	8,512	2,887	2,888	984
Vocabulary	1,172	34	-	-
OOV rate	-	-	3.2%	0.1%

Re-ranking Model

- The FST generates the most likely concept annotations.
- These are used to build annotation pairs, $\langle s^i, s^j \rangle$.
  - positive instances if s^i *more correct* than s^j,
- The trained binary classifier decides if s^i is more accurate than s^j.
- Each candidate annotation s^i is described by a word sequence where each word is followed by its concept annotation.

Re-ranking framework



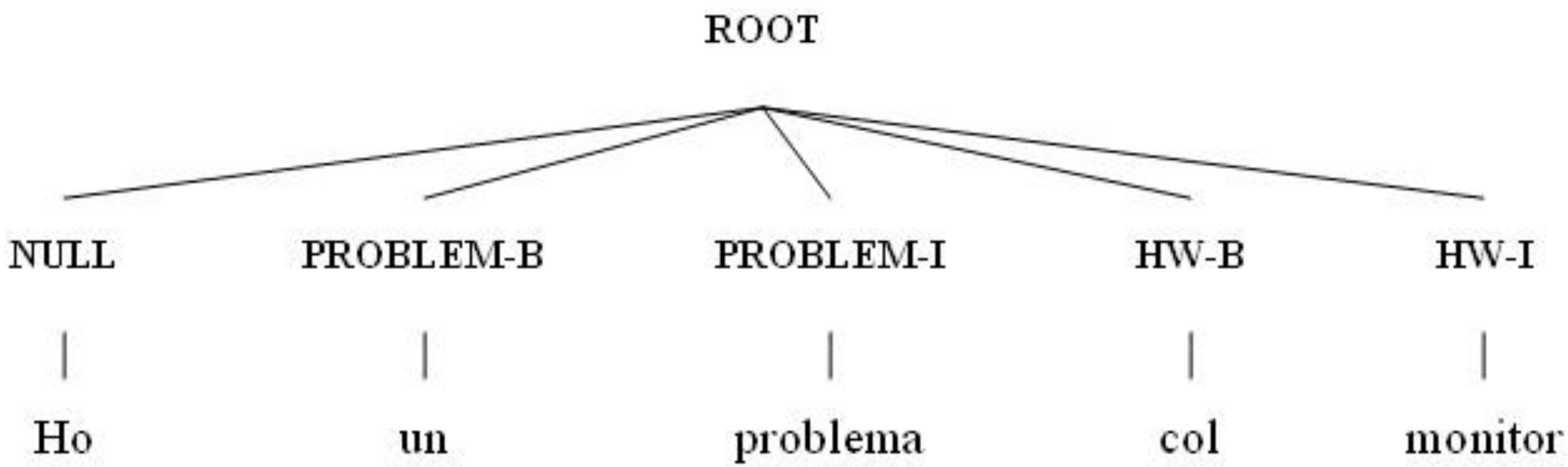
Example

- *I have a problem with the network card now*

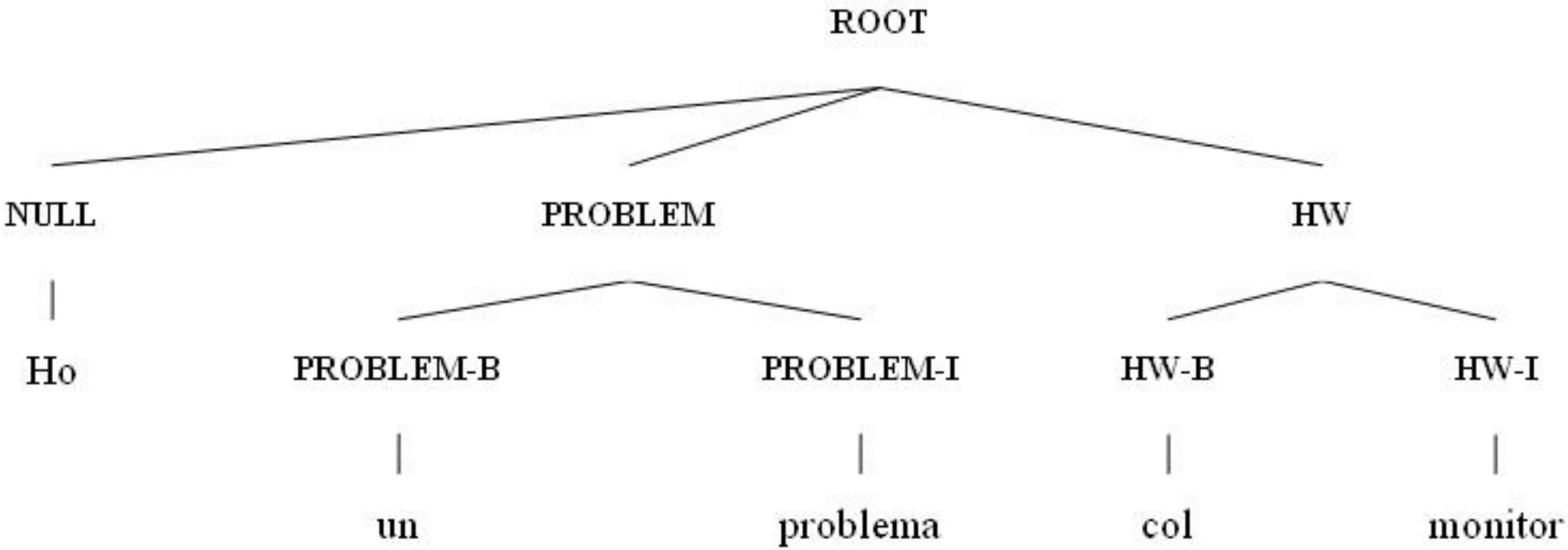
s^i: *I* **NULL** *have* **NULL** *a* **NULL** *problem* **PROBLEM-B** *with* **NULL** *my* **NULL** *monitor* **HW-B**

s^j: *I* **NULL** *have* **NULL** *a* **NULL** *problem* **HW-B** *with* **NULL** *my* **NULL** *monitor*

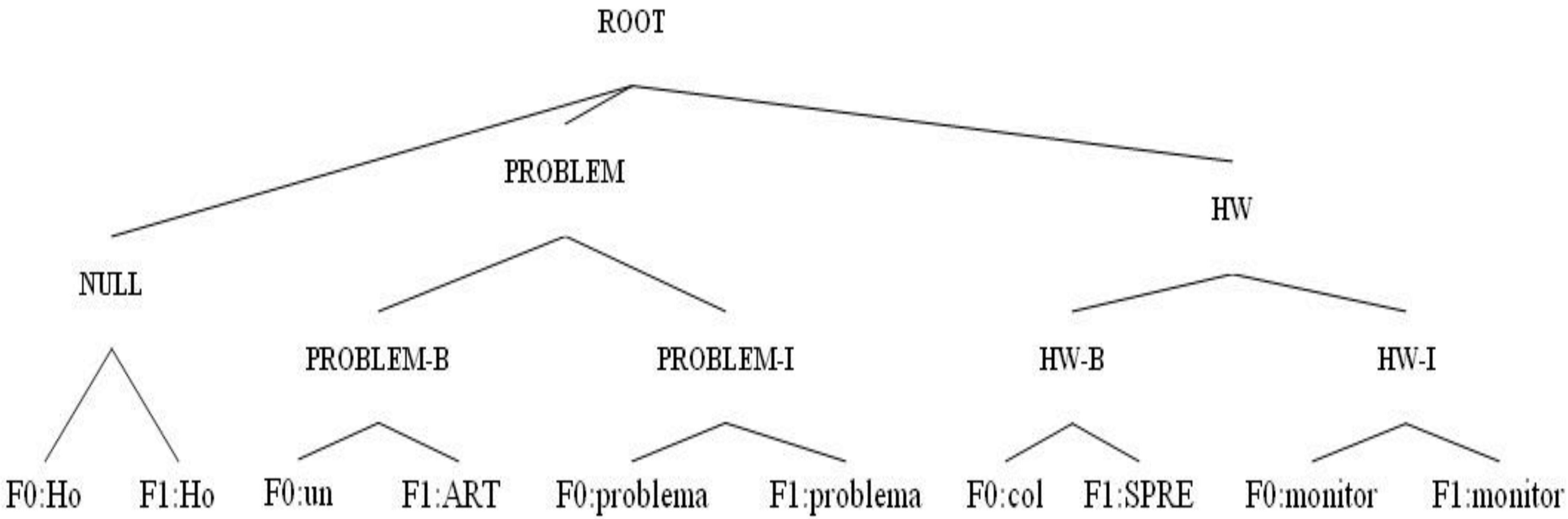
Flat tree representation



Multilevel Tree



Enriched Multilevel Tree



Results

Model	Concept Error Rate
SVMs	26.7
FSA	23.2
FSA+Re-Ranking	16.01

≈ 30% of error reduction of the best model

Structured Perceptron

Inputs: Training set (x_i, y_i) for $i = 1 \ldots n$

Initialization: $\mathbf{W} = 0$

Define: $F(x) = \mathrm{argmax}_{y \in \mathbf{GEN}(x)} \mathbf{\Phi}(x, y) \cdot \mathbf{W}$

Algorithm: For $t = 1 \ldots T, i = 1 \ldots n$

$z_i = F(x_i)$

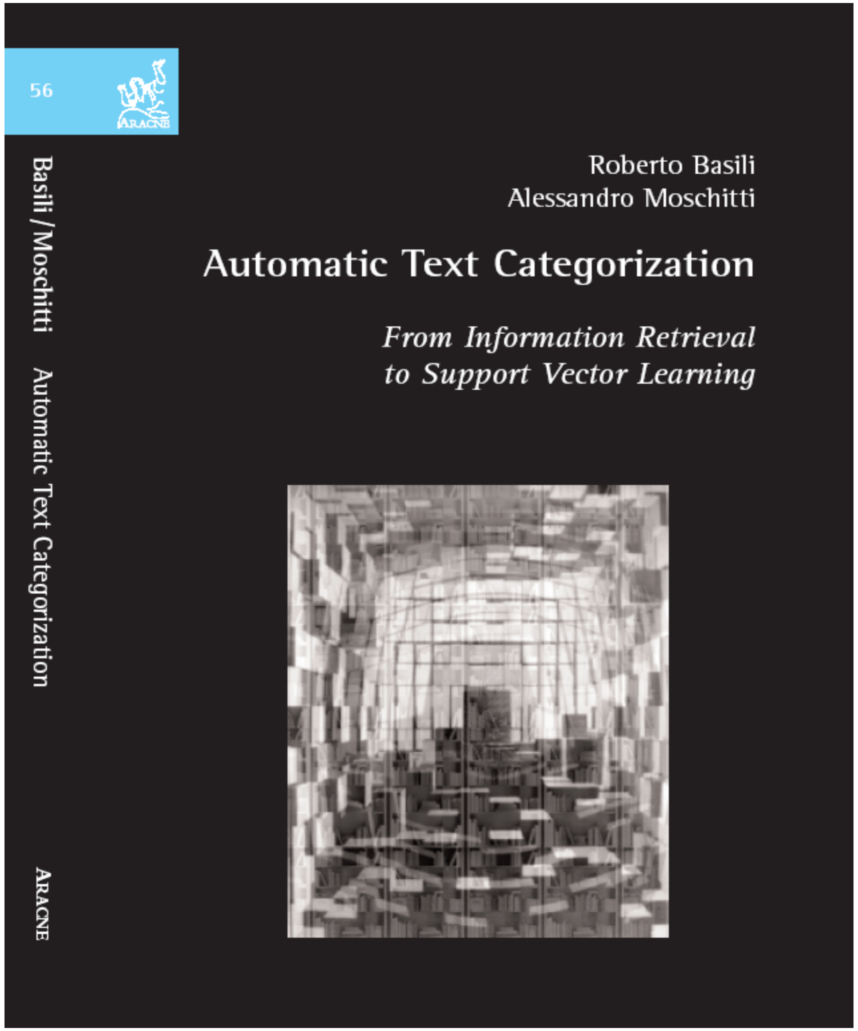
If $(z_i \neq y_i)$ $\quad \mathbf{W} = \mathbf{W} + \mathbf{\Phi}(x_i, y_i) - \mathbf{\Phi}(x_i, z_i)$

Output: Parameters $\mathbf{W}$

References

- Alessandro Moschitti, Silvia Quarteroni, Roberto Basili and Suresh Manandhar, *Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classification*, Proceedings of the 45th Conference of the Association for Computational Linguistics (ACL), Prague, June 2007.
- Alessandro Moschitti and Fabio Massimo Zanzotto, *Fast and Effective Kernels for Relational Learning from Texts*, Proceedings of The 24th Annual International Conference on Machine Learning (ICML 2007), Corvallis, OR, USA.
- Daniele Pighin, Alessandro Moschitti and Roberto Basili, *RTV: Tree Kernels for Thematic Role Classification*, Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-4), English Semantic Labeling, Prague, June 2007.
- Stephan Bloehdorn and Alessandro Moschitti, *Combined Syntactic and Semanitc Kernels for Text Classification*, to appear in the 29th European Conference on Information Retrieval (ECIR), April 2007, Rome, Italy.
- Fabio Aiolli, Giovanni Da San Martino, Alessandro Sperduti, and Alessandro Moschitti, *Efficient Kernel-based Learning for Trees*, to appear in the IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Honolulu, Hawaii, 2007

An introductory book on SVMs, Kernel methods and Text Categorization



References

- Roberto Basili and Alessandro Moschitti, *Automatic Text Categorization: from Information Retrieval to Support Vector Learning*, Aracne editrice, Rome, Italy.
- Alessandro Moschitti, *Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees*. In Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany, 2006.
- Alessandro Moschitti, Daniele Pighin, and Roberto Basili, *Tree Kernel Engineering for Proposition Re-ranking*, In Proceedings of Mining and Learning with Graphs (MLG 2006), Workshop held with ECML/PKDD 2006, Berlin, Germany, 2006.
- Elisa Cilia, Alessandro Moschitti, Sergio Ammendola, and Roberto Basili, *Structured Kernels for Automatic Detection of Protein Active Sites*. In Proceedings of Mining and Learning with Graphs (MLG 2006), Workshop held with ECML/PKDD 2006, Berlin, Germany, 2006.

References

- Fabio Massimo Zanzotto and Alessandro Moschitti, *Automatic learning of textual entailments with cross-pair similarities*. In Proceedings of COLING-ACL, Sydney, Australia, 2006.
- Alessandro Moschitti, *Making tree kernels practical for natural language learning*. In Proceedings of the Eleventh International Conference on European Association for Computational Linguistics, Trento, Italy, 2006.
- Alessandro Moschitti, Daniele Pighin and Roberto Basili. *Semantic Role Labeling via Tree Kernel joint inference*. In Proceedings of the 10th Conference on Computational Natural Language Learning, New York, USA, 2006.
- Alessandro Moschitti, Bonaventura Coppola, Daniele Pighin and Roberto Basili, *Semantic Tree Kernels to classify Predicate Argument Structures*. In Proceedings of the the 17th European Conference on Artificial Intelligence, Riva del Garda, Italy, 2006.

References

- Alessandro Moschitti and Roberto Basili, *A Tree Kernel approach to Question and Answer Classification in Question Answering Systems*. In Proceedings of the Conference on Language Resources and Evaluation, Genova, Italy, 2006.
- Ana-Maria Giuglea and Alessandro Moschitti, *Semantic Role Labeling via FrameNet, VerbNet and PropBank*. In Proceedings of the Joint 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL), Sydney, Australia, 2006.
- Roberto Basili, Marco Cammisa and Alessandro Moschitti, *Effective use of wordnet semantics via kernel-based learning*. In Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005), Ann Arbor(MI), USA, 2005

References

- Alessandro Moschitti, Ana-Maria Giuglea, Bonaventura Coppola and Roberto Basili. *Hierarchical Semantic Role Labeling*. In Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005 shared task), Ann Arbor(MI), USA, 2005.
- Roberto Basili, Marco Cammisa and Alessandro Moschitti, *A Semantic Kernel to classify texts with very few training examples*. In Proceedings of the Workshop on Learning in Web Search, at the 22nd International Conference on Machine Learning (ICML 2005), Bonn, Germany, 2005.
- Alessandro Moschitti, Bonaventura Coppola, Daniele Pighin and Roberto Basili. *Engineering of Syntactic Features for Shallow Semantic Parsing*. In Proceedings of the ACL05 Workshop on Feature Engineering for Machine Learning in Natural Language Processing, Ann Arbor (MI), USA, 2005.

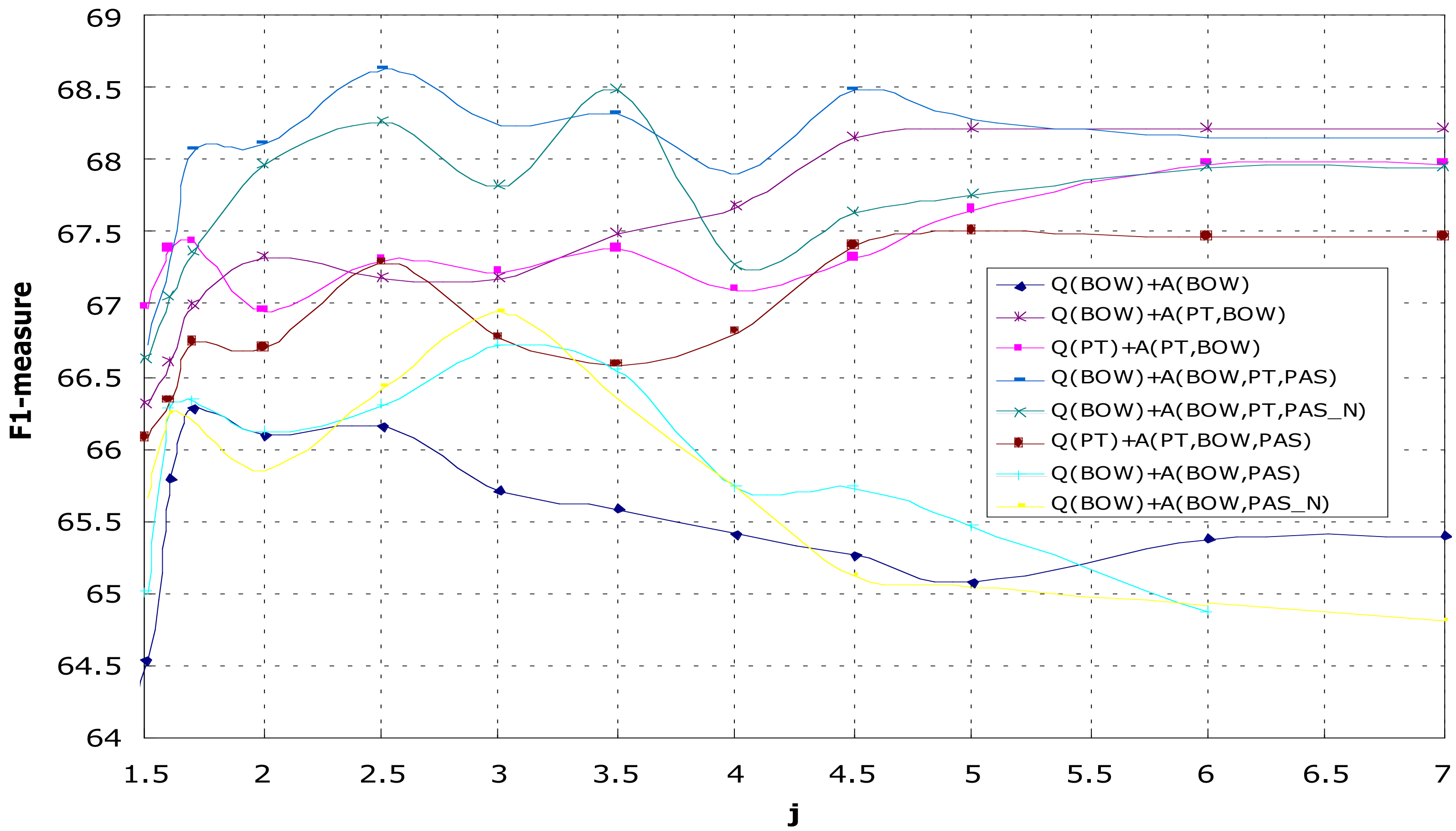
References

- Alessandro Moschitti, *A study on Convolution Kernel for Shallow Semantic Parsing*. In proceedings of ACL-2004, Spain, 2004.
- Alessandro Moschitti and Cosmin Adrian Bejan, *A Semantic Kernel for Predicate Argument Classification.* In proceedings of the CoNLL-2004, Boston, MA, USA, 2004.
- M. Collins and N. Duffy, *New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron*. In ACL02, 2002.
- S.V.N. Vishwanathan and A.J. Smola. *Fast kernels on strings and trees*. In Proceedings of Neural Information Processing Systems, 2002.

References

- *AN INTRODUCTION TO SUPPORT VECTOR MACHINES (and other kernel-based learning methods)*
  N. Cristianini and J. Shawe-Taylor Cambridge University Press
- Xavier Carreras and Llu´ıs M`arquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *proceedings of CoNLL’05*.
- Sameer Pradhan, Kadri Hacioglu, Valeri Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2005. Support vector learning for semantic argument classification. *to appear in Machine Learning Journal*.

The Impact of SSTK in Answer Classification



Mercer's conditions (1)

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

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

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

Def. B.13 *Positive (Semi-) definite Matrix*
A square matrix $\mathbf{A} \in \mathbb{R}^n \times \mathbb{R}^n$ *is said to be positive (semi-) definite if its eigenvalues are all positive (non-negative).*

Mercer's conditions (2)

Proposition 2.27 *(Mercer's conditions)*
Let X be a finite input space with $K(\vec{x},\vec{z})$ a symmetric function on X. Then $K(\vec{x},\vec{z})$ is a kernel function if and only if the matrix

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

is positive semi-definite (has non-negative eigenvalues).

- If the Gram matrix: $G = k(\vec{x}_i, \vec{x}_j)$
  is positive semi-definite there is a mapping ϕ that produces the target kernel function

The lexical semantic kernel is not always a kernel

- It may not be a kernel so we can use ***M´·M***, where ***M*** is the initial similarity matrix

Proposition B.14 *Let $\boldsymbol{A}$ be a symmetric matrix. Then A is positive (semi-) definite iff for any vector $\vec{x} \neq 0$*

$$\vec{x}'\boldsymbol{A}\vec{x} > \lambda\vec{x} \quad (\geq 0).$$

From the previous proposition it follows that: If we find a decomposition $\boldsymbol{A}$ in $M'M$, then $\boldsymbol{A}$ is semi-definite positive matrix as

$$\vec{x}'\boldsymbol{A}\vec{x} = \vec{x}'\boldsymbol{M}'\boldsymbol{M}\vec{x} = (\boldsymbol{M}\vec{x})'(\boldsymbol{M}\vec{x}) = \boldsymbol{M}\vec{x} \cdot \boldsymbol{M}\vec{x} = ||\boldsymbol{M}\vec{x}||^2 \geq 0.$$

Efficient Evaluation (1)

- In [Taylor and Cristianini, 2004 book], sequence kernels with weighted gaps are factorized with respect to different subsequence sizes.
- We treat children as sequences and apply the same theory

$$\Delta(n_1, n_2) = \mu\left(\lambda^2 + \sum_{p=1}^{lm} \Delta_p(c_{n_1}, c_{n_2})\right),$$

Given the two child sequences $s_1 a = c_{n_1}$ and $s_2 b = c_{n_2}$ (a and b are the last children), $\Delta_p(s_1 a, s_2 b) =$

$\mathbf{D_p}$

$$\Delta(a, b) \times \boxed{\sum_{i=1}^{|s_1|} \sum_{r=1}^{|s_2|} \lambda^{|s_1| - i + |s_2| - r} \times \Delta_{p-1}(s_1[1:i], s_2[1:r])}$$

Theory

- Kernel Trick
- Kernel Based Machines
- Basic Kernel Properties
- Kernel Types