Multi-Task Structured Prediction for Entity Analysis: Search Based Learning Algorithms

OSU Oregon State University

ACML 2017 Seoul, Korea

Chao Ma, †Janardhan Rao Doppa, Prasad Tadepalli, Hamed Shahbazi and Xiaoli Fern

Oregon State University, †Washington State University, USA

Motivation & Problem

- A NLP system usually solves multiple tasks;
- Some of these tasks are highly inter-dependent.

✓ We focus on three tasks for *Entity Analysis*

Named Entity Recognition | Coreference Resolution | Entity Linking

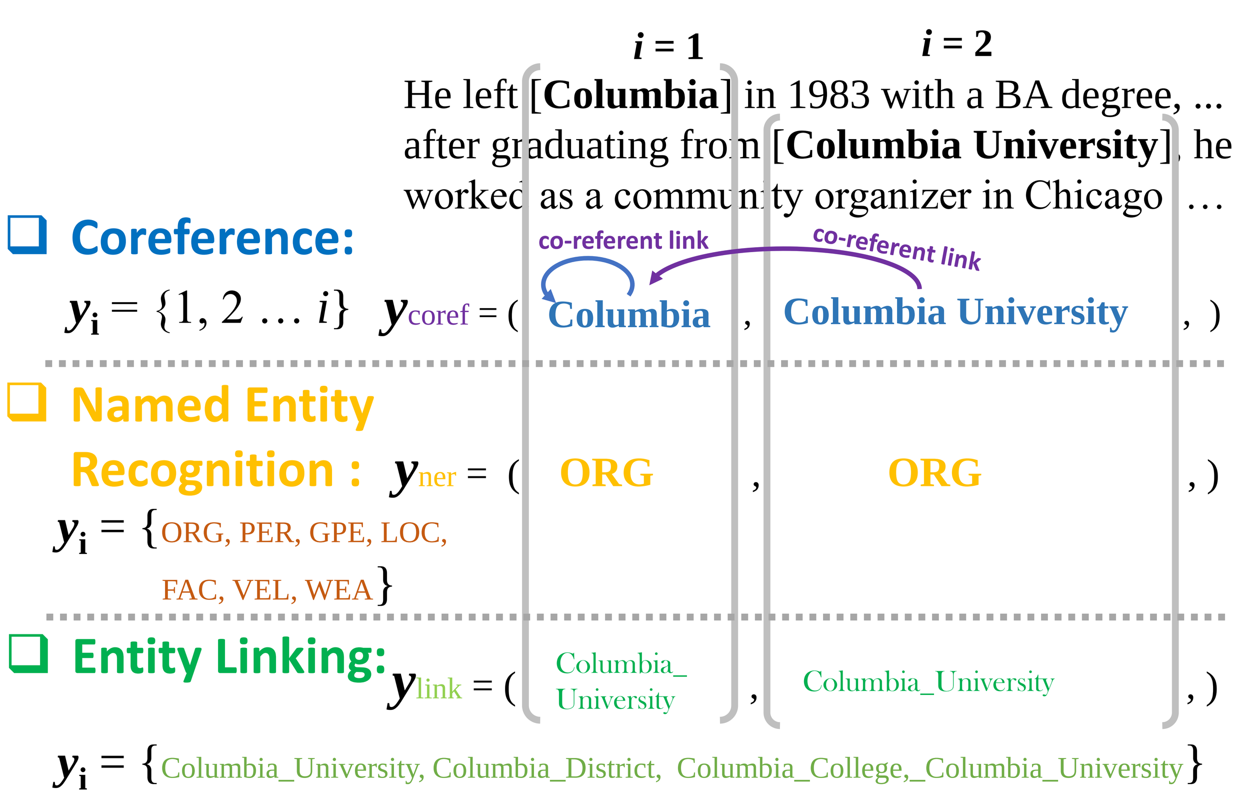
❑ Properties

- Each task predict a structural output;
- Task dependency has ***no ordering***.

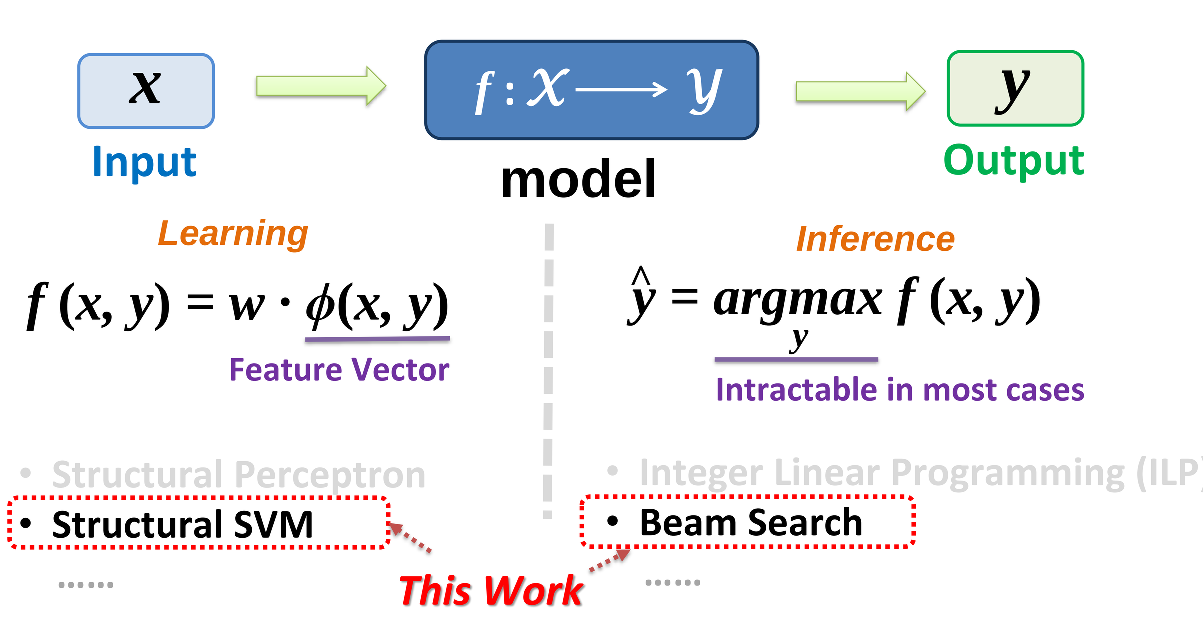
❑ Challenges

- How to exploit their interdependencies?
- How to control the error propagation?
- How to reduce the inference time complexity?

Tasks Introduction

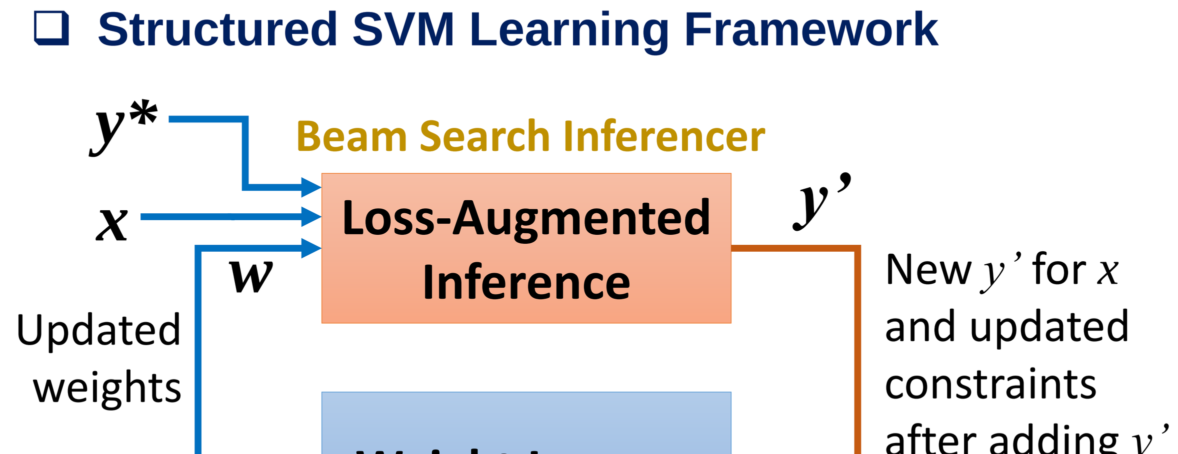


Single-Task Structured Prediction



SSVM Learning with Search-Based Inference

❑ **Structured SVM Learning Framework**



Three main advantages of search based inference

❑ Inference time complexity is not sensitive to the feature complexity;

This is fairly important in MTSP when using inter-task features, especially higher order features.

❑ Can optimize any arbitrary non-decomposable losses;

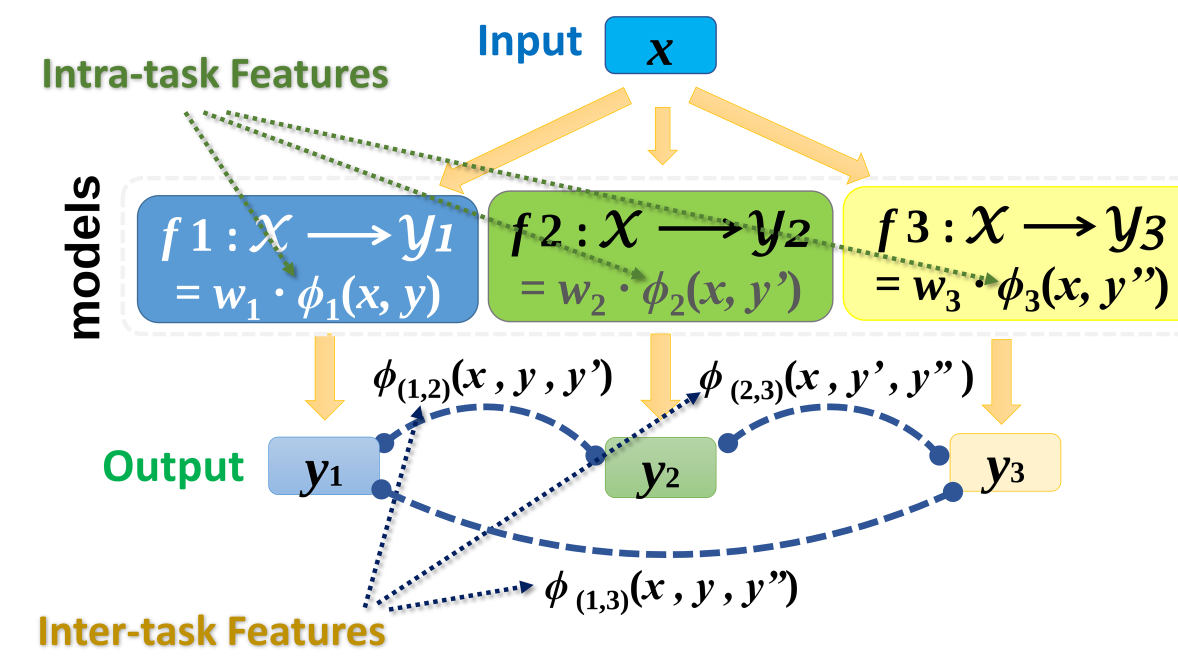
When doing loss-augmented inference in structured SVM:

- **Task specific losses, e.g., coreference CoNLL score;**
- **(Weighted) task compatibility losses, TAC-KBP NERLC score;**

❑ Can apply pruning to control the inference speed;

For example, ILP inference usually uses a *blackbox* solver, which is not flexible to apply pruning.

Multi-Task Structured Prediction



❑ **interdependencies between tasks are captured by the inter-task features.**

Inter-task Features for Entity Analysis

Coref-NER: e.g.: Agreement of NER tags of two coreferent mentions

ORG ≠ ORG

Coref-Link: e.g.: Relation of KB entries of two coreferent mentions

University is-same-category University

Mathematics is-sub-category Mathematics education

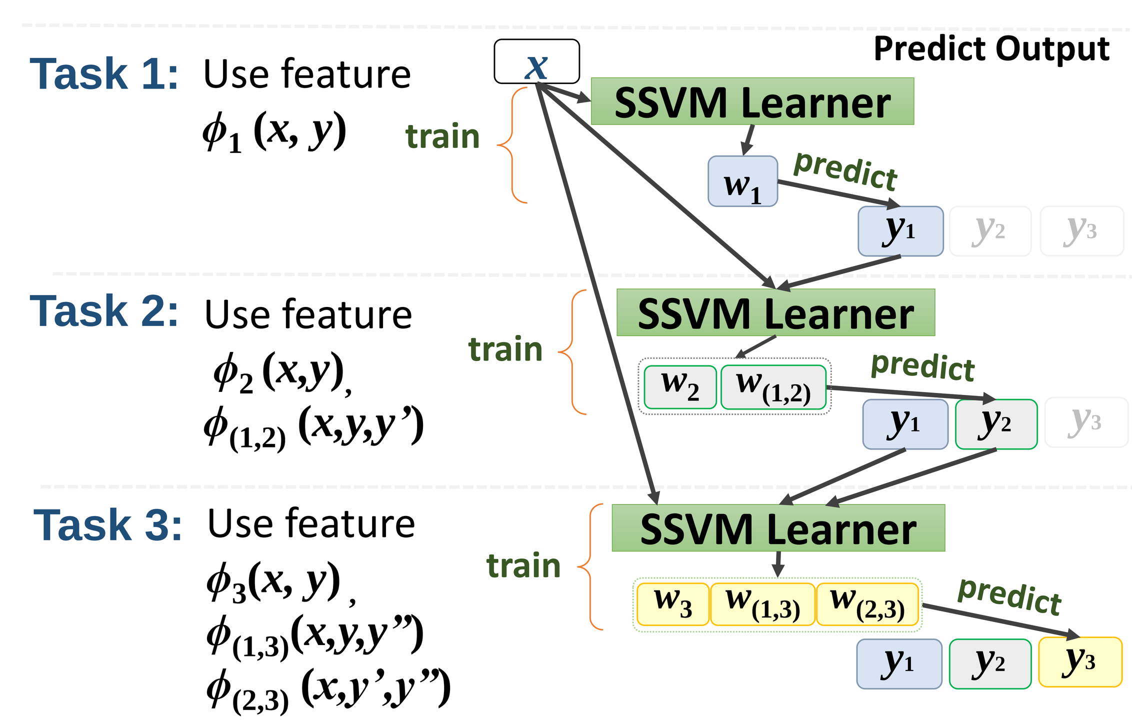
NER-Link: e.g.: NER-tag and Category pair indicator

(ORG, University)
(ORG, Institute)
(PER, President) ...

Bonus to the co-related pair

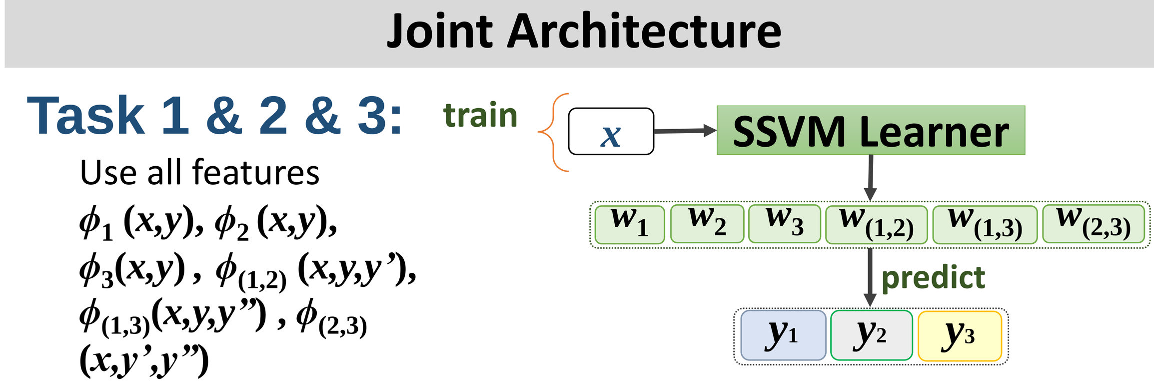
MTSP Architectures

Pipeline Architecture



❑ **Learning *k* (= 3) independent models, one after another.**

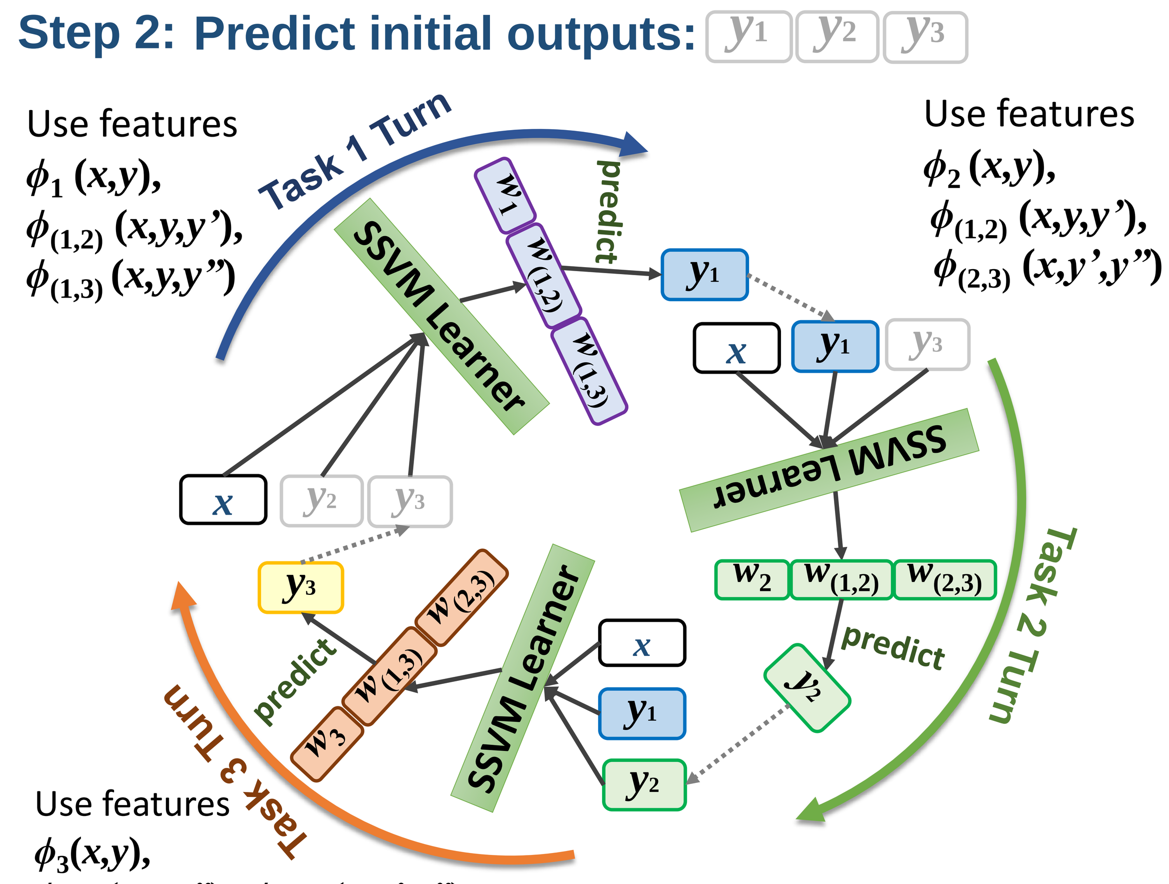
Joint Architecture



Cyclic Architecture

Step 1: Define a order: Task 1 → Task 2 → Task 3

Step 2: Predict initial outputs: y_1 y_2 y_3



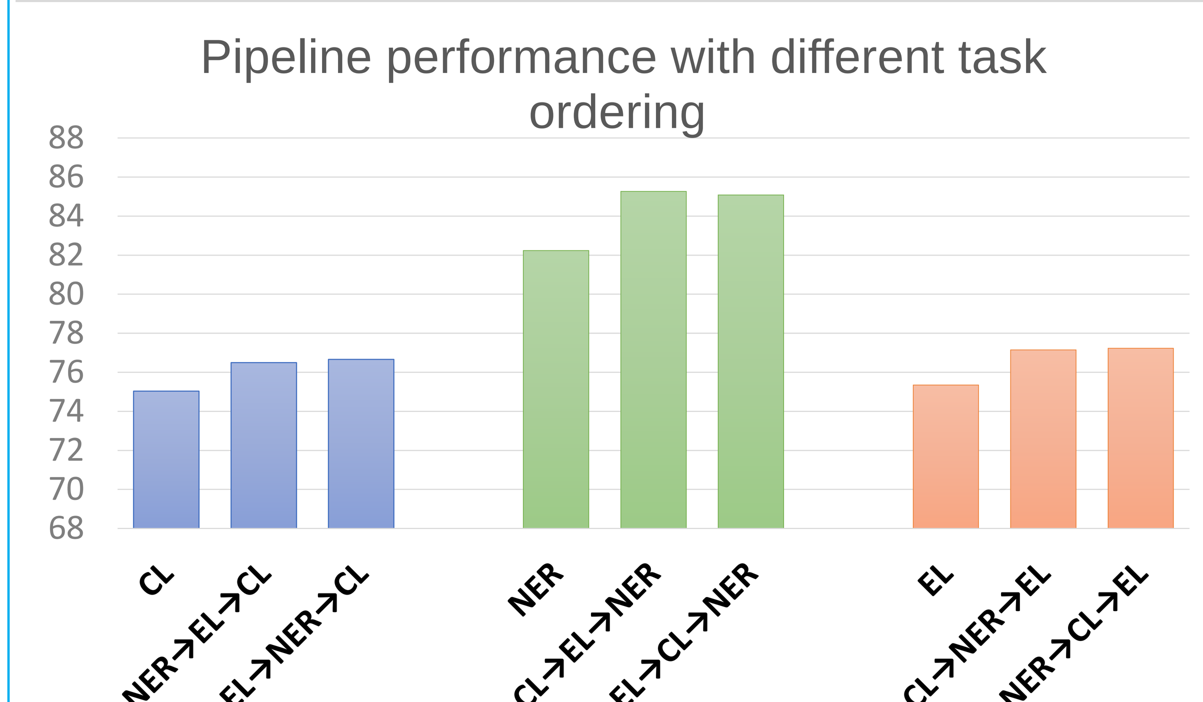
Experimental Results

Experimental Setups

	ACE 2005	TAC-KBP 2015
❑ Corpora:	(338/144/117 docs)	(132/36/167 docs)
❑ Knowledge Base:	Wikipedia (2014)	Freebase (2015)
❑ Evaluation:	Coref. CoNLL; NER Hamming; Link. Hamming	Cross.Coref. CEAFm & {NER, Link.} NERLC

✓ All metrics are accuracies, the larger the better

Pipeline Architecture Performance



- Each group of bars represents one task. In each group, we show the accuracy when the task is placed at first (1st bar), or at last (2nd and 3rd bar).

❑ The task performs better when it is placed last in order.

❑ There is no ordering that allows the pipeline to reach peak performance on all the three tasks.

Joint & Cyclic Architecture Performance

ACE05 Test Set Performance

Algorithms.	Coref	NER	Link	Train time
	CoNLL	*Accu.*	*Accu.*	*Min.*
Berkeley	76.35	85.60	76.78	31min
a. Results of Joint Architecture without Pruning				
STSP	75.04	82.24	75.36	9min
Joint w. Rand Init	75.35	82.20	76.99	48min
Joint w. Good init	77.58	85.71	78.77	34min
b. Results of Joint Architecture with Pruning				
Score-agnostic	77.07	85.63	78.71	16min
Score-sensitive	77.85	87.18	80.28	37min
c. Results of Cyclic Architecture				
Unshrd-Wt-Cyclic	77.29	84.18	80.67	11min
Shared-Wt-Cyclic	76.53	82.16	79.60	10min

❑ **Joint-Good-Init > STSP** Interdependency, captured by inter-task features, does benefit the system.

❑ **Joint-Good-Init > Joint-Rand-Init** Search-based inference for large SP problems suffers from local optima and can be mitigated by a good initialization.

❑ **Search-based MTSP is competitive or better** than the state-of-the-art systems.

❑ **Score-sensitive** pruning further improves the *joint* architecture. While **score-agnostic** pruning brings around 2 times speed up.

❑ **Unshared-Weight-Cyclic** performs better than **Shared-Weight-Cyclic,** but neither of them are sensitive to the task ordering.

Conclusions

1. Formulated the problem of multi-task structured prediction (MTSP) for entity analysis.
2. Applied the search-based learning framework, where structured SVM is employed for training and beam search for inference.
3. Develop the *cyclic* architecture that performs as good as *joint* architecture, but with a much faster speed.